THE UNIVERSITY *of* EDINBURGH

Unsupervised Induction of Semantic Roles

Joel Lang

Doctor of Philosophy

Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2011

Abstract

In recent years, a considerable amount of work has been devoted to the task of automatic frame-semantic analysis. Given the relative maturity of syntactic parsing technology, which is an important prerequisite, frame-semantic analysis represents a realistic next step towards broad-coverage natural language understanding and has been shown to benefit a range of natural language processing applications such as information extraction and question answering.

Due to the complexity which arises from variations in syntactic realization, data-driven models based on *supervised learning* have become the method of choice for this task. However, the reliance on large amounts of semantically labeled data which is costly to produce for every language, genre and domain, presents a major barrier to the widespread application of the supervised approach.

This thesis therefore develops *unsupervised* machine learning methods, which automatically induce frame-semantic representations without making use of semantically labeled data. If successful, unsupervised methods would render manual data annotation unnecessary and therefore greatly benefit the applicability of automatic frame-semantic analysis.

We focus on the problem of semantic role induction, in which all the argument instances occurring together with a specific predicate in a corpus are grouped into clusters according to their semantic role. Our hypothesis is that semantic roles can be induced without human supervision from a corpus of syntactically parsed sentences, by leveraging the syntactic relations conveyed through parse trees with lexical-semantic information.

We argue that semantic role induction can be guided by three linguistic principles. The first is the well-known constraint that semantic roles are unique within a particular frame. The second is that the arguments occurring in a specific syntactic position *within a specific linking* all bear the same semantic role. The third principle is that the (asymptotic) distribution over argument heads is the same for two clusters which represent the same semantic role.

We consider two approaches to semantic role induction based on two fundamentally different perspectives on the problem. Firstly, we develop feature-based probabilistic latent structure models which capture the statistical relationships that hold between the semantic role and other features of an argument instance. Secondly, we conceptualize role induction as the problem of partitioning a graph whose vertices represent argument instances and whose edges express similarities between these instances. The graph thus represents *all* the argument instances for a particular predicate occurring in the corpus. The similarities with respect to different features are represented on different edge layers and accordingly we develop algorithms for partitioning such multi-layer graphs.

We empirically validate our models and the principles they are based on and show that our graph partitioning models have several advantages over the feature-based models. In a series of experiments on both English and German the graph partitioning models outperform the feature-based models and yield significantly better scores over a strong baseline which directly identifies semantic roles with syntactic positions.

In sum, we demonstrate that relatively high-quality shallow semantic representations can be induced without human supervision and foreground a promising direction of future research aimed at overcoming the problem of acquiring large amounts of lexical-semantic knowledge.

Acknowledgements

First and foremost I would like to thank Mirella Lapata and Charles Sutton for their *exceptional* effort throughout the past years. This thesis and I personally have benefitted a lot from their engaging supervision which went beyond what one can normally expect. In addition to supporting me in the "day-to-day business" they managed to point me in good directions and helped paint the big picture. I am very fortunate to have worked with them.

I am furthermore grateful to all the people who through their valuable feedback have helped improve this thesis, in particular Sharon Goldwater, Amit Dubey, Phil Blunsom, Ivan Titov and Alex Klementiev. Likewise, I thank James Henderson, Paola Merlo, Nikhil Garg and Andrea Gesmundo for the various discussions, which were very helpful for preparing the final version of this thesis.

I would also like to express my sincerest thanks towards Ewan Klein and Lluís Màrquez for their time and effort. I very much appreciate that they have agreed to review this thesis.

A special thanks to Abby, Aciel, Ben, Christos, Dave, Des, Erich, Greg, John, Mark, Micha, Michael, Michal, Mike, Moreno, Saša, Silvia, Stella, Tom, Trevor, Yannis and Yansong for the pleasant times at ILCC.

Finally, I thank the EPSRC for generously funding this PhD work.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Joel Lang*)

Declaration of Previously Published Related Work

Some of the material presented in this thesis has been previously published. The basis of the work presented in Section 4.2 was published in Lang and Lapata (2010). The work in Section 3.3 and Chapter 5 is a refinement of the work published in Lang and Lapata (2011a) and Lang and Lapata (2011b).

Table of Contents

1 Introduction **1**

1.1 Frame Semantics 2

1.2 Frame-Semantic Analysis with Supervision 3

1.3 Unsupervised Frame Induction 5

1.3.1 Problem Definition 6

1.3.2 Characterizing the Unsupervised Setting 7

1.4 Thesis Outline 8

1.4.1 Hypothesis 8

1.4.2 Proposed Methods 9

1.4.3 Contributions 10

2 Frame Semantics **12**

2.1 Frames and Semantic Roles 13

2.1.1 Frames 13

2.1.2 Semantic Roles 14

2.1.3 Frames and Semantic Roles at the Clausal Level 15

2.1.4 Linguistic Perspective on Inferring Semantic Roles 17

2.1.5 Non-Clausal Frames 22

2.2 Empirical Resources 23

2.2.1 FrameNet 24

2.2.2 PropBank 25

2.3 Frame-based Language Understanding 26

2.3.1 Frame-Semantic Analysis: Task Definition 27

2.3.2 State of the Art 27

2.3.3 Frame Semantics and Reasoning 33

2.3.4 Applications 34

2.4 Summary . . . 37

3 Problem Setting **38**

3.1 Problem Formulation . . . 39
3.2 Data . . . 40
3.3 Predicate and Argument Identification . . . 41
3.4 Evaluation . . . 44
3.5 Baseline Method for Semantic Role Induction . . . 46
3.5.1 Baseline Evaluation . . . 49
3.6 Summary . . . 49

4 Feature-based Probabilistic Models **52**

4.1 Semantic Roles as Latent Variables . . . 53
4.1.1 Models . . . 53
4.1.2 Results and Analysis . . . 61
4.2 Semantic Roles as Canonical Syntactic Positions . . . 64
4.2.1 Standard Linkings and Canonical Syntactic Positions . . . 64
4.2.2 Logistic Classifier with Latent Variables . . . 65
4.2.3 Results and Analysis . . . 69
4.3 Related Work . . . 73
4.4 Summary . . . 75

5 Role Induction via Similarity-Graph Partitioning **77**

5.1 Measuring Similarity . . . 78
5.1.1 Similarity Functions . . . 79
5.2 Graph Partitioning . . . 80
5.2.1 Graph Construction . . . 81
5.2.2 Problem Formulation . . . 82
5.3 Agglomerative Graph Partitioning . . . 83
5.3.1 Runtime Analysis . . . 90
5.4 Graph Partitioning by Label Propagation . . . 91
5.4.1 Runtime Analysis . . . 95
5.4.2 Comparison with Agglomerative Clustering . . . 96
5.5 Results and Analysis . . . 96
5.6 Eager Similarity Combination . . . 98
5.6.1 Label Propagation on Single-Layer Graphs . . . 101

5.6.2 Combining Feature Similarities Heuristically 102
5.6.3 Learning Instance Similarities from Data 104
5.6.4 Results and Analysis 105
5.7 Related Work 107
5.8 Summary 108

6 Semantic Role Induction for German **110**
6.1 Word Order and Case Marking in German 111
6.2 The SALSA Dataset 114
6.3 Experimental Setup 116
6.4 Results and Analysis 118
6.5 Summary 120

7 Conclusions **122**
7.1 Contributions 122
7.2 Future Work 124

Bibliography **128**

A Significance Testing **139**

B Argument Identification Rules **140**

C Label Sets **141**

D Sample Output **149**

Chapter 1

Introduction

Frame semantics (Fillmore, 1968; Minsky, 1974) is a formalism, which has proven useful for building language understanding systems. In frame semantics, the meaning of a natural language sentence such as

(1.1) Carl repaired the motor within a week.

is analyzed by identifying the situation described by the sentence, here conveyed by the predicate *Repair*, and the entities participating in the situation, here *Carl*, *the motor* and *a week*. Furthermore each entity is characterized in terms of a *semantic role*, which describes the way it is involved in the situation. For example, *Carl* can be characterized as the entity instigating the action, i.e., the *Agent* of the action (see Figure 1.1).

In recent years, a considerable amount of work has been devoted to the task of automatically computing a frame-semantic analysis for a given input sentence. Due to the complexity which arises from variations in the syntactic realization of semantic roles, data-driven models based on supervised learning have become the method of choice for this task. Unfortunately, the reliance on large amounts of semantically labeled data which is costly to produce for every language, genre and domain, presents a major barrier to the widespread application of the supervised approach. This thesis therefore develops *unsupervised* learning methods, which automatically induce frame-semantic

representations without making use of semantically labeled data.

Unsupervised methods offer a promising but also challenging alternative. If successful, such methods would render manual data annotation unnecessary, thereby benefitting the coverage and portability of frame-based language understanding systems. This thesis takes a step in this direction and shows that it is indeed possible to induce relatively high-quality representations without resorting to supervised learning or human-constructed semantic resources.

In the following, we will first introduce the main ideas behind frame semantics in Section 1.1[1] and then, in Section 1.2, define the problem of frame-semantic analysis and describe existing approaches based on supervised learning. Then, in Section 1.3, we will motivate, define and characterize the problem of inducing frame-semantic representations without supervision. Finally, in Section 1.4 we outline the methods developed in this thesis and describe our contributions.

1.1 Frame Semantics

Frame semantics was originally developed by Fillmore (1968) as a formalism for analyzing clausal semantics and independently by Minsky (1974) as a framework for knowledge representation. *Frames* are structures, which represent arbitrary situations such as *eating something*, *a court trial* or *an election campaign*. A frame specifies the concomitants of a situation, in particular the entities that participate in the situation. The way in which a particular entity is involved in the situation is characterized by a *semantic role*, which thus captures the abstract, prototypical relationship between the entity and the situation. For example, for frames representing actions such as *eating*, *breaking* or *repairing* the *Agent* role designates the instigating entity of that action and the *Patient* role designates the entity which is affected by the action (see Figure 1.1).

Frames can represent situations of arbitrary granularity (elementary or complex) and

[1] The discussion here is short and a more detailed introduction will be given in Chapter 2

accordingly frame-semantic analysis can be conducted on linguistic units of varying sizes, e.g. phrases (e.g., Meyers et al., 2004), sentences (e.g., Fillmore, 1968) or whole documents (e.g., Minsky, 1974), but most work has been devoted to frame semantics as a formalism for sentence-level semantic analysis and most commonly it has been applied for the analysis of verbal predicate-argument structures, in accordance with classical linguistics which has ever since emphasized the centrality of the verb and its function in conveying atomic semantic propositions in the form of clauses (Fillmore, 1968). Figure 1.1 shows a frame representation together with several possible syntactic realizations.

Frames can be viewed as an intermediary representation between syntax and semantics. While the representation abstracts away from a particular surface-level syntactic configuration, it is still intimately tied to the surface form. For example frame entities are not grounded and constructs such as quantifiers or logical connectives are left uninterpreted and not present in the semantic representation. In this sense, the representation is *shallow* and less expressive than other representations such as a full first-order logical form, but also less difficult to compute.

This relative simplicity has contributed to the success of frame semantics as a practical approach to language understanding, especially for open domains where full logic-based systems often fail to produce an analysis, i.e., suffer from low coverage. Indeed, automatically computed frame-semantic analyses like the one given Figure 1.1 have been shown to benefit a variety of applications ranging from information extraction (Surdeanu et al., 2003) and question answering (Shen and Lapata, 2007), to machine translation (Wu and Fung, 2009) and summarization (Melli et al., 2005).

1.2 Frame-Semantic Analysis with Supervision

The bulk of previous work has based frame-semantic analysis on supervised learning, as initiated by Gildea and Jurafsky (2002) and promoted by a range of shared tasks (Carreras and Màrquez, 2004; Litkowski, 2004; Carreras and Màrquez, 2005; Baker et al., 2007; Surdeanu et al., 2008; Hajič et al., 2009).

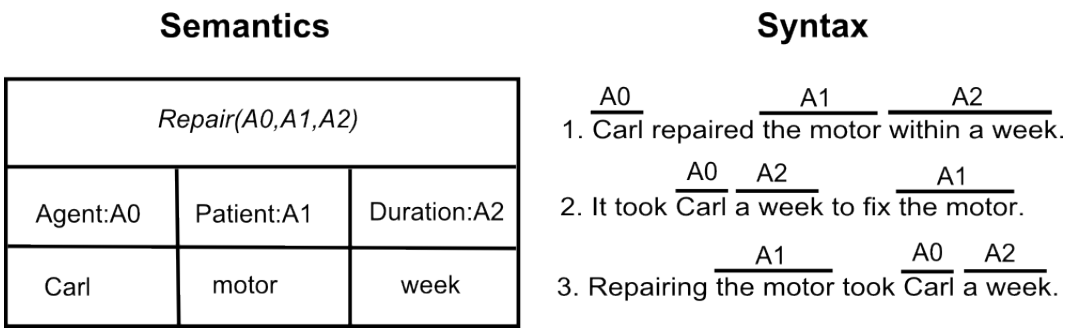


Figure 1.1: Example of a frame for the action predicate *Repair* and several possible syntactic realizations. The frame specifies the participating entities aka arguments as well as their semantic roles. Here, *Carl* is the instigating entity, i.e., *Agent*, and *motor* is the entity affected by the action, i.e., the *Patient*. Additionally the frame specifies temporal information that is assigned a semantic role *Duration*.

The goal of extracting frame instantiations from a given input sentence is typically formulated as a three-step problem (see Màrquez et al., 2008):

1. Predicate identification: identifying the verbal predicates that occur in the sentence (e.g., *Repair* in Sentence 1 of Figure 1.1);

2. Argument identification: identifying the arguments of each predicate (e.g., *Carl*, *the motor* and *within a week* in Sentence 1 of Figure 1.1);

3. Argument classification: labeling each argument with a semantic role (e.g., *Agent*, *Patient* and *Duration* for the arguments in Sentence 1 of Figure 1.1).

Steps (1) and (2) can be viewed as binary classification problems, which require making a decision about the status of a particular unit (word or phrase) in the input sentence. Step (1) decides whether a unit is a predicate or not and Step (2) whether it is an argument of a particular predicate. Step (3) is again a classification problem in which argument units assigned a semantic role.

Given this breakdown into three cascaded classification problems, a natural solution is

to train a classifier for each step, which maps input units onto outputs, i.e., either binary yes/no-decisions for Steps (1) and (2) or semantic roles for Step (3). Supervised classification is a well-studied problem in machine learning, and engineering classifiers for Steps (1)-(3) mainly involves determining the set of features which inform the classification decision. State-of-the-art systems typically employ further, more complex mechanisms, in particular such which account for interdependencies between the classification Steps (1)-(3) and such for achieving an optimal joint classification of all frame entities (details will be described in Chapter 2).

The classifiers are learned using supervision from a corpus of labeled data, in which each sentence is paired with gold standard output. Thus, although the approach is conceptually simple, in practice it entails a large data labeling effort to create the training corpus. This motivates the use of unsupervised methods developed in this thesis and introduced in the following section.

1.3 Unsupervised Frame Induction

The obvious drawback of conceptualizing frame-semantic analysis as a supervised learning problem is that building a broad-coverage system requires prohibitively large amounts of human-labeled data, due to the fact that the syntactic realization of semantic roles is irregular across verbs and often tied to lexical idiosyncrasies. Therefore training an open-domain system requires a sufficiently large training sample for each of the thousands of verbs that may occur. Consequently the data labeling effort for broad-coverage resources like PropBank (Palmer et al., 2005) and FrameNet (Ruppenhofer et al., 2006) amounts to multi-million US-Dollar expenditures, which of course prohibits the application of the supervised learning approach to a wider range of genres and languages. This raises the question investigated in this thesis: do unsupervised methods offer a viable alternative to supervised methods?

Beyond the immediate motivation of reducing the data requirements for broad-coverage frame-based language understanding, this thesis also constitutes part of a more general effort to build unsupervised systems for natural language processing. The solution

to many problems, for example parsing, to date still relies extensively on supervised learning, with implications similar to those described above for frame-semantic analysis. In some cases, supervised methods also tend to replace ingenuity and a theoretical understanding of the problem at hand with intransparent, data-driven models ('black boxes'), which is questionable, at least from a research perspective. Consequently, unsupervised methods have received much attention in recent years, leading to increasingly accurate unsupervised models for various tasks such as part-of-speech tagging (see Christodoulopoulos et al., 2010) or parsing (Klein, 2005; Seginer, 2007; Snyder et al., 2009, i.a.). With this thesis we contribute to the general effort in unsupervised learning, which ultimately should benefit the coverage and portability of many kinds of natural language processing systems and yield better insights into the problems involved.

1.3.1 Problem Definition

The goal of computing a frame-semantic analysis is the same irrespective of the learning paradigm, i.e., supervised vs. unsupervised, namely to extract frame instantiations from a given input sentence. In the unsupervised setting we will refer to the problem as *frame induction*. We assume that the input is syntactically analyzed in the form of a dependency tree, thereby isolating frame induction from syntactic parsing. Reducing the data requirements for parsing is certainly also an important concern, but outside the scope of this thesis. The choice of a dependency representation as opposed to a constituent representation simplifies various aspects of the task, for example argument identification, but is not imperative in the sense that all of the models developed in this thesis could also be formulated on the basis of a constituent representation.

Along general lines, the problem of frame induction can be stated in the same way as in the supervised case (see Section 1.2). We thus adopt the three-step decomposition for the unsupervised setting. Predicate identification (Step 1) remains the same. We slightly reformulate argument identification (Step 2) as the task of discarding as many non-semantic arguments as possible without discarding actual semantic arguments. This means that the argument identification component does not make a final

positive decision for any of the candidate units[2]; rather, a final decision is only made in the subsequent argument classification (Step 3), which differs fundamentally from the supervised setting. Since in the unsupervised setting there is no predefined set of semantic roles, these must be induced from the data itself and we will refer to this problem as *role induction*. Role induction follows the contract of a clustering problem in which the units selected by Step (2) are grouped into clusters representing semantic roles. Each induced cluster can then be given an interpretation and a label which is applied to all the units contained in the cluster. Clusters that do not represent any semantic role (containing non-argument units) can be labeled accordingly.

1.3.2 Characterizing the Unsupervised Setting

Unsupervised learning is known to be challenging for many real-world problems, e.g., parsing (Klein, 2005) and frame induction is no exception. This section gives reasons why this is the case, most of which apply to other problems as well. The main qualitative difference to the supervised setting is of course the lack of an extensional definition of the target concepts, i.e., for role induction a set of examples for each possible semantic role. Therefore, inductive reasoning is applicable to a less extent than in the supervised setting and reasoning must instead rely more on prior knowledge about the problem. The challenge of unsupervised learning thus consists in finding a strong inductive bias (see Gordon and Desjardins, 1995) based on this prior knowledge, which will guide the induction process towards the correct target concept.

More technically speaking, the unsupervised setting makes it harder to define a learning objective function, whose optimization will yield an accurate model. In the supervised setting, the objective function can directly reflect training error, i.e., some quantification of the mismatch between model output and the gold standard. Thereby, the model can be trained to replicate human output for a given input under mathematical guarantees regarding the accuracy of the trained model. Whatever objective function we come up with in the unsupervised setting, it is difficult to guarantee that it will result in a model that is (roughly) as accurate as a human, even if the optimization

[2]Some supervised systems have previously defined argument identification in the same way, e.g. Koomen et al. (2005).

problem itself is well understood.

It is also more difficult to incorporate rich feature sets into an unsupervised model (see Berg-Kirkpatrick et al., 2010). Unless we explicitly know exactly how features interact, more features will not lead to a more accurate model, contrariwise they may even decrease performance. For the supervised setting, there exist methods such as support vector machines (Cortes and Vapnik, 1995) with which the feature interactions relevant for a particular learning task can to large extent be determined automatically and thus a large number of features can be included even if their significance is not clear a priori. This contrasts with the unsupervised setting where feature-rich models are difficult to implement even where they would have good theoretical justification and would lead to improvements under different learning conditions.

A further complication of unsupervised learning concerns evaluation and the development process itself. A quantitative evaluation is normally conducted against a gold standard test set. Thus at least here, we cannot avoid using labeled data. Moreover, the predefined gold standard will not reflect previously unknown data characteristics which are discovered by the unsupervised method and thus evaluation scores will only assess the extent to which the induced representations coincide with predefined notions. Finally, in practice, model development is an iterative, trial-and-error process which is difficult to conduct without a labeled dataset (either development or test set) that can be used to assess the current model. Thus, while unsupervised methods do not require labeled data per se, they may in practice not manage to completely supersede it.

1.4 Thesis Outline

1.4.1 Hypothesis

The hypothesis underlying this thesis is that semantic roles can be induced without human supervision from a corpus of syntactically parsed sentences, by leveraging the syntactic relations conveyed through parse trees with lexical-semantic information. We

claim that by combining both syntactic and lexical information it is possible to build models which induce semantic roles more accurately than models which rely solely on syntactic information. We hypothesize that in principle these two sources of information (syntactic parses and lexical-sematic information) are sufficient to induce high-quality frame-semantic representations.

1.4.2 Proposed Methods

This thesis will focus on role induction, i.e., the problem of grouping candidate verbal arguments into clusters representing semantic roles. Chapter 3 will show that predicate and argument identification can be conducted through a set of relatively simple rules which rely exclusively on analyzing the *syntactic* structure of the input sentence. In contrast, the role induction problem is more challenging and must be informed by both syntactic and lexical-semantic cues.

We will propose and compare two fundamentally different approaches to role induction. In Chapter 4 we model semantic roles through two classes of feature-based probabilistic latent structure models. In the first model class the semantic role is directly modeled as a latent variable, whose value indicates the particular role of the argument. Thus, given the argument's observed features, we can determine its semantic role by inferring the value of the latent semantic role variable. In the second model class, a layer of latent variables implements a generalization mechanism that abstracts away from an argument's observed syntactic position to its (unobserved) semantic role, relying on the fact that there is a close correspondence between the two. Our evaluation and analysis will reveal that it is difficult to develop a well-performing model with this approach. None of our feature-based probabilistic models manages to consistently outperform a baseline which identifies semantic role with syntactic positions.

In Chapter 5 we take a fundamentally different approach to role induction, that relies on judgements regarding the similarity of argument instances with respect to their semantic roles. Rather than modeling the probabilistic relationships between argument features, we model when two argument instances have the same role or have

differing roles. Given such similarity judgements our data is naturally modeled as a graph, whose vertices correspond to argument instances and whose edge weights express similarities. Based on this representation, we conceptualize role induction as a graph partitioning problem, in which the goal is to partition the graph into clusters of vertices representing semantic roles. We demonstrate that this approach manages to significantly increase the quality of induced clusters over the baseline.

In Chapter 6 we test the graph partitioning approach on German, in order to exemplify its applicability to languages other than English and its robustness with respect to variations of the underlying syntactic representation. We show that results for German are qualitatively similar to English, confirming the cross-lingual applicability of the models and the principles they are built on.

1.4.3 Contributions

The main contributions of this thesis are as follows.

1. We develop and compare three different conceptualizations of the role induction problem: (a) as probabilistic inference in a latent-variable model; (b) as determining the *canonical* syntactic position of an argument; and (c) as a graph partitioning problem. Conceptualizations (a) and (b) correspond to the feature-based approach mentioned in the previous section, whereas (c) corresponds to the fundamentally different *similarity-driven* approach.

2. We formulate of a set of principles that serve as a theoretically sound basis for building language-independent role induction models. The first is the well-known constraint that semantic roles are unique within a particular frame. The second is that the arguments occurring in a specific syntactic position *within a specific linking* all bear the same semantic role. The third principle is that the (asymptotic) distribution over argument heads is the same for two clusters which represent the same semantic role. We empirically validate the models and the principles through a set of experiments on both English and German.

3. We devise new general-purpose models for classification and clustering. In the context of the conceptualization described under 1(b) we develop a variant of the logistic classifier, in which a layer of latent variables mediates between the input variables and the target variable in order to improve generalization. In the context of 1(c) we develop *multi-layer* similarity graph partitioning methods for inferring semantic role clusters, which is a novel extension of established single-layer graph partitioning methods.

4. We contribute to the body of work on *similarity-driven* models, by demonstrating their suitability w.r.t. modeling our problem, their effectiveness, and their computational efficiency. The comparison with feature-based models reveals several advantages of the similarity-driven models and thereby provides a complementary view to much contemporary research which has concentrated on and argued in favor of feature-based models.

5. We identify and analyze major difficulties such as lexical sparsity which arise, yielding insights which contribute towards developing better frame-based language understanding systems that are less reliant on labeled data.

6. We foreground a promising direction of research aimed at inducing shallow semantic representations without human supervision, which is a logical step given the relative maturity of syntactic parsing technology and the difficulty of overcoming the lexical-semantic bottleneck (Padó, 2007), i.e., the problem of acquiring large amounts of lexical-semantic knowledge.

Chapter 2

Frame Semantics

One of the most interesting questions regarding how language is used to convey knowledge concerns the transition from semantics to syntax: how exactly are semantic representations mapped onto surface-level forms and vice-versa? Frame semantics is a formalism which bridges this gap between language in its syntactic form and the underlying knowledge structures which it expresses. It provides both a theoretical model of language understanding as well as a practical methodology for building language understanding systems.

This chapter provides an overview of frame semantics, covering a breadth of issues: theoretical, practical, linguistic, ontological, resources, implementations, etc. The material is presented in three parts. The first part introduces the basic concepts and terminology. The second part describes FrameNet and PropBank, two large-scale frame semantic resources. Finally, we discuss the how frame semantics can be used as a formalism for building language understanding systems.

2.1 Frames and Semantic Roles

The two most important concepts in frame semantics are those of a *frame* and a *role*. A frame represents a particular situation and its concomitants, including participating entities. A role characterizes how a participating entity is involved in a situation. The next two sections describe these two concepts in detail, and the two following sections will then move on to describe how frames and roles are expressed in language.

2.1.1 Frames

Minsky (1974) introduced frames as "a data-structure for representing a stereotyped situation" (p. 1). Frames represent arbitrary situations: *eating at a table*, *a court trial*, *an election campaign*, etc. Minsky conceived frames as pieces of knowledge which help understand specific instances of the situations they describe. In order to fulfil this purpose, frames are accompanied by information about involved entities, temporal information, causal information, and so on. For example, an *Election Campaign* frame might specify entities such as a *Candidate* and the *Function* he or she is running for. Frames were conceived as an alternative framework for knowledge representation, moving away from unstructured logic-oriented approaches that tried "to represent knowledge as collections of separate, simple fragments"(Minsky, 1974, p. 1). Under Minsky's framework, knowledge is organized into a system of interrelated and inter-referring frames. The *Election Campaign* frame for example, could be related to an *Election* subframe, which describes the details of the election day including for example the implications of winning or losing an election, and so on.

Before Minsky, Fillmore (1968) came up with the notion of a *case frame*. In contrast to Minsky, Fillmore's frames are just as much linguistic as they are ontological. Case frames are structures holding together the arguments bound to a particular predicate. The classical theory assumes a verbal predicate and nominal elements, as in the following example (see also Figure 1.1).

(2.1) Carl repaired the motor within a week.

The three nominal elements *Carl*, *the motor* and *a week* are bound together (related) by the verbal predicate *Repair*, with which they form a case frame. Each element stands in a particular prototypical relationship to the predicate (frame). For example, the case frame for *Repair* often specifies the entity instigating the action, i.e., the *Agent* of the action (here *Carl*). Such prototypical relationships are also called *semantic roles* and will be discussed in the following sections. In terms of transformational grammar (Chomsky, 1965), case frames constitute the deep structure of what is realized as a clause on the surface. While they specify the clausal elements, including possible lexical choices, case frames do not as such contain syntactic information, e.g., about the ordering of these elements. By definition, they are intimately tied to the linguistic units they serve to represent. Compared to Minsky's frames, they therefore tend to represent more elementary pieces of knowledge; single actions, states, events or processes rather than complex situations.

A third strang of work was put forward by Abelson and Schank (1977). Their *scripts* are frame-like structures, particularly aimed at capturing frequently recurring situations and modeling the behavior of the interacting participants. The famous *restaurant script*, for example, maintains schematic knowledge of what happens when a person visits a restaurant (e.g., *sitting down at a table* or *ordering food*). Scripts comprise whole stories, which are typically communicated over multiple sentences, and therefore aim at discourse-level understanding.

2.1.2 Semantic Roles

Much like frames represent prototypical situations, *semantic roles* represent prototypical relationships that characterize how a participating entity is involved in a situation. A common role, used to describe the instigator of an action, is the *Agent*. It applies to arbitrary situations in which some participant is causing the world to change (see Example 1 in the previous section). Other common examples are *Patient*, the participant affected by an action, *Instrument*, the entity used to perform an action, *Location*, the

place where an action takes place, etc. The choice of semantic roles and their granularity may depend on the specific domain of discourse and the particular application at hand. The following paragraphs will contrast the two basic options of using a small set of general roles vs. a large set of specific roles.

Fillmore (1968) developed a role system comprising *Agent*, *Patient*, *Instrument*, *Location*, *Result* (what results from an action or event) and a role *Neutral*, whose semantics are determined by the particular verbal predicate. These roles served the purposes of linguistic analysis, rather than knowledge representation and Fillmore called them *semantic cases*, in analogy to grammatical cases such as *Nominative*, *Accusative*, *Dative*, etc. In his *case grammar*, discussed in Section 2.1.3, he relates semantic to grammatical cases, accounting for various morpho-syntactic phenomena in terms of the underlying deep structure.

Fillmore (1968)'s roles are general enough to characterize the semantic arguments of arbitrary predicates, in other words, their scope of application is universal. Universality is important from a linguistic standpoint, because it leads to a concise linguistic theory, but the question which roles to include in such a universal role set has been disputed (Dowty, 1991). Which roles are necessary and sufficient? How general or specific should roles be? What roles are present in all languages?

The alternative to universal roles are situation-specific roles such as *Buyer* and *Seller* occurring together with the predicate *Buy* (see Figure 2.2 for an illustration of the frame *Buy*). Such situation-specific roles have a preciser meaning and thus support more detailed reasoning about situations, however at the cost of increased complexity and loss of generality. In Section 2.2.1 we will discuss FrameNet (Ruppenhofer et al., 2006), a large-scale lexical resource comprising many such situation-specific roles.

2.1.3 Frames and Semantic Roles at the Clausal Level

This section describes how frames and roles are mapped onto clauses consisting of a verbal predicate and one or several arguments, which corresponds to the classical scope

of frame-semantic analysis (Fillmore, 1968). Section 2.1.5 will describe the realization of frames across different linguistic units, for example frames that are expressed over multiple sentences, or frames that are bound to nominal rather than verbal predicates.

In classical linguistics clauses are understood as units expressing elementary semantic propositions (Fillmore, 1968). The verbal predicate expresses an action, event, state or process, or generally speaking, some kind of relationship between its arguments, which represent the entities that are involved. Fillmore (1968) proposed to analyze predicate-argument structures in terms of a system of semantic roles ('*semantic cases*'), which characterize how an argument relates semantically to the predicate. His original role system consisted of six roles: *Agent*, *Patient*, *Instrument*, *Location*, *Result* and *Neutral* (see Section 2.1.2 for an explanation of these roles). He noticed that the configurations of roles occurring together with a predicate were not arbitrary, but rather fixed and regular. For example, the verb *open* always takes a mandatory *Patient* and is optionally accompanied by an *Agent* and/or an *Instrument*. Thus, the configuration for *open* can be represented as a frame structure $Patient + (Instrument) + (Agent)$ which lists the mandatory and optional cases. Such 'case frames' serve as typological semantic representations of a predicate and provide a basis for explaining the surface-level syntactic realization of the predicate's arguments.

Specifically, Fillmore studied how semantic cases are linked to the grammatical cases that are present at the syntactic level, giving an account for various morpho-syntactic phenomena including case marking of nominals, grammatical function and subcategorization. While in English, grammatical cases (e.g. *Nominative*, *Accusative*, *Dative*, etc.) occur only in very limited form (e.g. with pronouns), morphologically richer languages such as the Slavic languages possess an elaborate system of grammatical cases, which are indicative of the underlying semantic case. Fillmore viewed the linking between semantics and syntax as resulting from a process in which semantic roles are iteratively chosen for filling a particular syntactic position (*Subject*, *Object*, *Prepositional Phrase* or *Adverbial*) according to a *selectional hierarchy*, specifying precedence. For example, *Agent* has precedence over the other roles for filling the *Subject* position under active tense, i.e., if *Agent* is present in the case frame then it will be mapped onto *Subject* position. Analogously, hierarchies were defined for other positions and under different conditions, e.g., for the passive voice.

In general, any generative grammar which works with semantic roles must integrate the concept of a *linking*, i.e., the mapping from semantic roles onto syntactic positions (see for example Levin (1986) who developed a linking theory for Kaplan and Bresnan (1982)'s lexical functional grammar). As we will discuss in Section 2.1.4.1 syntactic variation on the surface level can be interpreted as the result of differing underlying linkings. Such variation of verbal argument realization has also been called *alternation*.

Another important phenomenon coupled to semantic roles is *lexical selection*. Fillmore (1968) proposed that a particular semantic role imposes constraints on the class of possible lexical fillers. An obvious example of such a *selectional constraint* is that the entity taking the *Agent* role is animate, or otherwise capable of instigating an action. Since not all lexical units express such entities, the fillers of *Agent* therefore belong to a restricted subset. More recent work such as Resnik (1993) has addressed lexical selection at the syntactic level, by characterizing which types of words fill a particular syntactic position. But this is only appropriate to the degree that syntactic positions correspond directly to semantic roles. Lexical selection should in the first instance be seen as a semantic phenomenon.

2.1.4 Linguistic Perspective on Inferring Semantic Roles

This section conveys some important linguistic notions, which can serve as a theoretical basis for developing both supervised and unsupervised models that automatically infer the semantic roles of arguments. We will firstly describe how information about semantic roles is contained in both syntax *and* the lexical content of an argument. Second, we elaborate on the notions of linking and alternation from the previous section. The discussion will prove useful towards conceiving unsupervised models in the later chapters. Supervised models which implement the notions conveyed here will be discussed in Section 2.3.2.

2.1.4.0.1 Syntactic vs. Lexical Information Much information about the semantic role of a verbal argument is encoded in the syntactic structure of a sentence. The following examples give an idea of how syntactic features inform argument classification.

(2.2) $[\text{The cook}]_{SBJ}$ sliced the mushrooms.

The syntactic position of an argument is particularly indicative of its semantic role. For example, the *Subject* position often realizes the *Agent* of an action predicate, as in the sentence above. In fact, a feature which encodes the argument's position in the syntax tree by itself provides sufficient information for a good approximate role assignment. We will see this in Section 3.5, where we present a baseline role induction method which classifies arguments according to their syntactic position. Of course, exactly those cases which can be considered 'interesting', namely cases of syntactic variation in argument realization, will not be treated correctly by such a baseline. In order to account for such variation, *clause-level* information proves particularly important. For example, the verb voice feature informs about passivization:

(2.3) The mushrooms $[\text{were sliced}]_{PASS}$ by the cook.

While it is well understood how the verb voice feature influences the syntactic positioning of arguments, this is not necessarily the case for other features, where the interaction can be complex and specific to a particular class of verbs. An example of such a feature is the syntactic frame, which helps disambiguate between different senses of the same verb:

(2.4) $[\text{Food prices increased.}]_{INTRANS}$

(2.5) $[\text{The committee increased food prices.}]_{TRANS}$

In the example, the intransitive sentence (2.4) employs the non-agentive meaning of *increase*, but intransitivity is of course not generally an indication that the verbal predicate in non-agentive: a counter example is given by the sentence *The audience applauded*. Moreover, while we can typically provide a reasonable analysis for a few,

simple examples like the preceding one, it is important to keep in mind that in general it would be both difficult and laborious to come up with a set of rules for all verbs which would accurately describe all possible syntactic encodings of semantic roles.

An alternative source of information about the semantic role of an argument is its lexical content (see Zapirain et al., 2010). For example, the following lexical units can be easily combined into semantically plausible sentences:

(2.6) *eats*, *Michael*, *a sandwich*

(2.7) *chased*, *the rat*, *the cat*

(2.8) *hired*, *the mogul*, *the bank*

Moreover, for both sentences (2.6) and (2.7) humans will most likely agree in how semantic roles are to be assigned to lexical units, because certain assignments will lead to implausible sentences such as *a sandwich eats Michael* or *the rat chased the cat*. For sentence (2.8) the assignment is more flexible, because of an ambiguity that cannot be resolved at the lexical level (*banks* and *moguls* can both hire each other). Nevertheless, in cases like (2.6) and (2.7) an assignment can be made based on lexical knowledge.

In fact we can make the (simplifying) assumption that *a particular content word is associated with a single semantic role for each predicate*. In analogy to the *one sense per context* heuristic (Yarowsky, 1995) that is often used in word sense disambiguation, we can refer to the assumption as the *one role per context* assumption. This conveys the view that role-semantic information is quasi-attached to the lexical units themselves, rather than arising only in the context of a particular construction. This is analogous to a distinction made in German syntax between *lexical* and *structural* case marking (S. Müller, 2007).

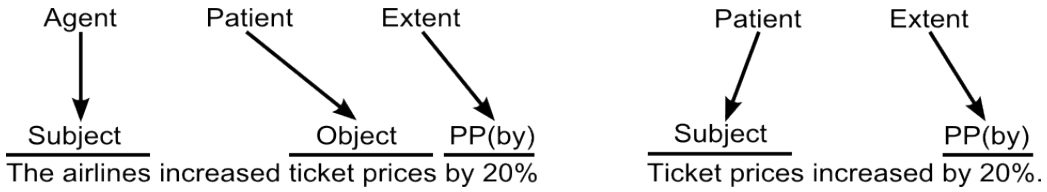


Figure 2.1: Illustration of two different linkings for the verb *increase*. The *Patient* role is once found in *Object* position and once in *Subject* position. Such variations in argument realization are known as alternations.

2.1.4.1 Linkings and Alternations

Much of the complexity in assigning semantic roles arises because of variation in the syntactic realization of arguments, a phenomenon known as *alternation*. There are a large number of alternation patterns, many of which are often characteristic for a particular group of verbs and closely related to the semantics of the predicate. Levin (1993) has conducted an extensive study of alternation phenomena and to give an idea, we borrow some examples from her.

Passive Alternation (p. 86, Nr. 306)

(2.9) The cook sliced the mushrooms.

(2.10) The mushrooms were sliced by the cook.

Induced Action Alternation (p.31, Nr. 24)

(2.11) Sylvia jumped the horse over a fence.

(2.12) The horse jumped over a fence.

Container Subject Alternation (p.82, Nr. 286)

(2.13) I incorporated the new results into the paper.

(2.14) The paper incorporates the new results.

There-Insertion (p.89, Nr. 322)

(2.15) A ship appeared on the horizon.

(2.16) There appeared a ship on the horizon.

A helpful notion for thinking about alternations like these is that of a *linking*. A linking specifies how arguments are mapped onto syntactic positions or, in the terminology of transformational grammar (Chomsky, 1965), it determines the correspondence between deep and surface structure. Formally, we can think of a linking as a mapping between semantic roles and syntactic positions of the arguments in a clause. For an example, consider Figure 2.1, which illustrates two different linkings for the verb *increase*. The first one corresponds to the mapping $\{Agent \mapsto Subject, Patient \mapsto Object, Extent \mapsto Prep(By)\}$ and the second one to the mapping $\{Patient \mapsto Subject, Extent \mapsto Prep(By)\}$, which differs with respect to the mapping of *Patient* and in that it does not define a mapping for *Agent*.

Alternations can be explained by the fact that verbs can be used together with different linkings. When two instances of the same verbal predicate use different linkings, corresponding syntactic positions may hold arguments with differing semantic roles. The variation in syntactic realization we observe is thus a result of varying underlying linkings. In the example given in Figure 2.1, the *Patient* role is mapped into *Object* position under the first linking, while it appears in *Subject* position under the second linking.

Conceptually speaking, if we knew the linking underlying a clause we could attempt to reconstruct the semantic role of each argument by inverting the mapping (this of

course only works if the particular syntactic position is associated with at most one semantic role). Unfortunately, the specific linking of a clause is not directly observed. Nevertheless, together with some additional assumptions, the notions conveyed in this section will lead to an approach which revolves around establishing a correspondence between semantic roles and the typical syntactic positions they occur in (see Section 4.2).

2.1.5 Non-Clausal Frames

The frames discussed so far, in particular Fillmore's case frames, are intimately bound to single clauses. However, under Minsky (1974)'s notion that frames can represent arbitrary simple or complex situations, there is no justification for this restriction. On one hand, language offers a multitude of constructions for expressing essentially the same semantic proposition. We can write

(2.17) Jim bought a donut for one pound.

(2.18) Jim bought a donut. It cost one pound.

(2.19) Jim's purchase of a donut cost him one pound.

The frame associated with these formulations should (or at least could) be the same, but only the first sentence uses a single clausal construction, whereas the second uses two clauses and the third uses a nominal predicate to express the *Buying* event. Moreover, there are complex situations that cannot be reasonably expressed within a single clause and frames representing such a situation have to be expressed over multiple clauses.

Some previous work has been devoted to nominal predicates and their arguments (Meyers et al., 2004). Real-world language is full of nominal predicate-argument constructions, that are realized for example via support verbs (e.g., *to make a decision*) or prepositional phrases (e.g., *election for president*). In fact, through the nominalization of verbs language provides a means for systematically translating verbal into nominal

constructions, whose argument realization patterns can be just as complex as for verbal predicates.

Less has been said about the principles underlying the linguistic realization of frames over multiple sentences. No mainstream *linguistic* theory of discourse such as Mann and Thompson (1988)'s rhetorical structure theory or Grosz et al. (1995)'s centering theory has adopted frames and roles explicitly into its analysis and Kamp and Reyle (1993)'s discourse representation theory addresses discourse semantics with first-order predicate logic, rather than frame semantics.

Authors such as Minsky (1974) and Abelson and Schank (1977) did explicitly connect discourse-level language understanding with frame-based representations. They however never elaborated the linguistic part of their theories. Arguably, the linguistic principles we are looking for would likely be less strict than those guiding sentence-level realization, much like discourse structure is less rigid than sentence-level syntax. Notable steps in this direction have recently been taken by various authors such as Gerber and Chai (2010) and Chambers and Jurafsky (2009) which we will discuss in Section 2.3.2.0.5. The growing interest in this type of frame-semantic analysis is also demonstrated by the Semeval 2010 task on linking events and their participants in discourse (Ruppenhofer et al., 2010).

2.2 Empirical Resources

Empirical resources are a key element for developing data-driven models for frame-semantic analysis as described in Section 2.3. Besides providing the basis for training supervised models they also enable the empirical linguistic study of frame semantics. This section describes FrameNet and PropBank, two large-scale role-semantic resources for English, each of which holds a large number of frames and accompanying semantic roles and documents their possible syntactic realization.

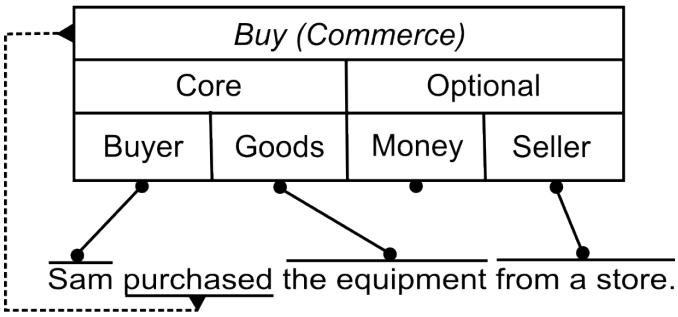


Figure 2.2: Example of the FrameNet frame *Buy (Commerce)*, which includes roles such as *Buyer*, *Goods*, *Money* and *Seller*. FrameNet distinguishes between core roles, which are necessarily present in every instantiation of the frame and non-core role whose instantiation is optional.

2.2.1 FrameNet

FrameNet (Ruppenhofer et al., 2006) is a lexical resource that associates words with meaning representations in the form of frames. Each word in the lexicon is a *frame evoking element*, i.e., a word used to express a particular frame, or several frames in the case of polysemous words. Each frame specifies the semantic roles of mandatory and optional entities and is accompanied with example sentences documenting the syntactic realization of these roles through annotations which mark the predicate and its arguments and indicate their semantic roles and grammatical functions. Further annotations include phrase types, syntactic features (e.g., occurrence within a relative clause), named entity labels, and so on. An example of a FrameNet frame is shown in Figure 2.2.

FrameNet also organizes frames into an ontology via relations such as *inheritance* (one situation is a special case of another), *perspectivation* (two frames describe the same situation from different perspectives), *composition* (one situation contains the other) and *temporal precedence* (one situation happens before the other).

FrameNet has annotated text excerpts from the British National Corpus, the Ameri-

can National Corpus and a corpus of newswire texts. As a follow-up on the discussion regarding non-clausal frames given in Section 2.1.5 it may be interesting to note that only sentences whose semantic roles are realized within the maximal projection of the predicate are annotated. Excepted from this rule are raising and control constructions as well as relative clauses but otherwise, frames spanning multiple clauses are not annotated. The current version 1.5 of FrameNet contains around 960 frames, around 11,600 predicates and around 150,000 annotated frame instantiations. Despite of development costs of around 5 Million US dollars, FrameNet does not (yet) include frames for all verbs and for many frames often contains insufficient training data for learning a reliable model.

2.2.2 PropBank

PropBank (Palmer et al., 2005) is a verb lexicon that associates verbs with frames. Each predicate is associated with one frame, which captures the possible configurations of semantic roles. While PropBank shares with FrameNet the basic idea of representing the meaning of predicate words via frame structures, the specific inventory of semantic roles differs largely from FrameNet. Most importantly, many of the semantic roles have a verb-specific meaning and are therefore only applicable to the arguments of one particular verb. More precisely, PropBank distinguishes between *core* and *adjunct* roles, where the adjunct roles (e.g. *Location*, *Extent* or *Time*), as their name indicates, are realized as adjuncts and can participate in any predicate's frame and are therefore defined globally for all predicates. In contrast, core roles are realized as verbal complements and defined individually for each verb, without relating them across verbs. Note that in contrast to FrameNet core roles, PropBank core roles are not necessarily present in the syntactic realization of a frame.

The syntactic realization of each frame is documented via exemplary sentences for which the predicate and its arguments are annotated. Core roles are simply labeled with an identifier[1] such as A0, A1, etc., in accordance with the fact that their interpretation is specific to the particular predicate. A special role *AA* is used for the rare case where an agent induces an action, e.g., *Sylvia jumped the horse over a fence.*,

[1] Here we use a simplified notation, i.e. A0 instead of Arg0, TMP instead of AM-TMP, and so on.

where *Sylvia* is the agent that is causing the horse to jump but not jumping herself. Adjunct arguments are labeled with one of the following eight roles: *Location* (LOC), *Extent* (EXT), *Cause* (CAU), *Time* (TMP), *Purpose* (PRC), *Manner* (MAN), *Direction* (DIR) and a general purpose *Adverbial* (ADV) role. Also annotated are reciprocal expressions (REC), predicatives (PRD), discourse connectives (DIS), negation (NEG) and modal verbs (MOD), although these are not semantic roles. PropBank was built as an extra annotation layer over the Wall Street Journal portion of the Penn Treebank (Marcus et al., 1993), and contains around 110,000 annotated frame instantiations. The sentences involve around 3,300 verbs and 4,500 predicates (verb senses). Development costs were similar to FrameNet, i.e., in the range of several million US Dollars.

2.3 Frame-based Language Understanding

Pioneering work in the 1970's and 1980's (e.g., Charniak, 1978) was mainly concerned with *domain-specific* systems[2] for language understanding and language generation. To date frames remain in use for such systems (e.g., Miller et al., 1996), where they are sometimes referred to as *templates*. Depending on the complexity of the discourse domain, developing the frames and specifying their possible syntactic realization requires a more or less time-intense knowledge-engineering effort, which has to be repeatedly invested for each new domain, genre and language.

In contrast to the early work, current systems commonly employ data-driven models rather than hand-coded rules for analyzing the input sentences. This shifts the engineering effort away from hand-coding rules over to labeling large amounts of data, which tends to require less expertise but just as much time. Like for other NLP problems, data-driven models are often better at handling the complexity of 'real-world' language data, which is rich in ambiguity, variation and lexical idiosyncrasies. In the remainder of this section we will therefore focus on the *empirical* approach to frame semantics, which reliant on resources like the ones described in the previous section has led to data-driven models for automatically extracting frames and labeling arguments with semantic roles. Gildea and Jurafsky (2002) popularized this idea and coined the

[2] See Levin (1977) for early work on open-domain frame-based language understanding.

term *semantic role labeling* to describe the frame-semantic analysis task. Their work was followed by a bulk of research on the task and its applications to open-domain language understanding. In the following we will firstly define semantic role labeling and then summarize the state of the art. In Section 2.3.4 we will then briefly discuss applications. Throughout this thesis we will follow other authors (e.g., Padó and Lapata, 2009) and refer to the task as frame-semantic analysis.

2.3.1 Frame-Semantic Analysis: Task Definition

This section briefly repeats the definition given in Section 1.2. The task of computing a frame-semantic analysis consists of extracting frame instantiations from a given input sentence and comprises three steps:

1. Identifying the verbal predicates that occur in the sentence (predicate identification);

2. Identifying the arguments of each predicate (argument identification);

3. Labeling each argument with a semantic role (argument classification).

The output of the task are frame instantiations, i.e., structures which reference a particular predicate and specify a set of entity-expressing units from the input sentence together with their semantic roles.

2.3.2 State of the Art

As described in Section 1.2 predicate identification, argument identification and argument classification can all be viewed as classification problems, which require making a decision about the status or class of a particular unit (word or phrase) in the input sentence. Standard systems (e.g., Johansson and Nugues, 2008) therefore commonly

execute a cascade of classifiers, which during the development phase have been trained with supervision from a corpus of labeled data. Since Gildea and Jurafsky (2002)'s seminal work, advances have mainly taken place in three areas: *classifiers*, *feature engineering* and *global optimization*. We will summarize these areas in the following and for a more complete treatment refer to Màrquez et al. (2008). As this thesis aims at relieving the data requirements we will also discuss previous work on *semi-supervised* learning, in which the training data consists of both labeled and unlabeled data. Unsupervised models will be discussed in the following chapters as related work.

2.3.2.0.1 Syntactic Analysis While syntactic analysis is not part of the core task of frame-semantic analysis defined in Section 2.3.1 it is an important prerequisite and influences the subsequent processing steps as well as the quality of the produced analyses significantly. Moreover, frame semantics is closely tied to syntax, which motivates conceiving models for joint syntactic and frame-semantic parsing. This is currently an active area of research which we will discuss in Section 2.3.2.0.4. In this section we will however treat syntactic parsing as an independent task, whose output constitutes the input to the frame-semantic analysis task. Many state-of-the-art systems make use of this sequential composition of syntactic parsing and frame-semantic analysis into a pipeline.

Both dependency- and constituent-based syntactic representations have been chosen as a basis for frame-semantic analysis. Initially, constituent representations were more common, while more recently dependency representations have gained popularity, as is reflected by the fact that the CoNLL 2005 Shared Task on Semantic Role Labeling (Carreras and Màrquez, 2005) was constituent-based, whereas the CoNLL 2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies (Surdeanu et al., 2008) was dependency-based. While the choice does not affect the general architecture and design rationale of the system, it does imply changes in terms of the specific processing that is involved. Importantly, argument identification on a dependency representation is simpler, since it only requires identifying the argument head word, as opposed to the exact boundaries of the whole constituent. Similarly, argument candidate pruning, the process of identifying parse tree nodes that potentially represent arguments, is commonly implemented via parse tree traversal algorithms which necessarily differ for the two types of representations; and of course the specific features

extracted from the parse tree are different.

Regardless of the type of representation, the syntactic parse tree provides the basis for implementing the classification steps that arise from the task formulation given in Section 2.3.1: classification decisions are made for parse tree nodes, which represent units from the input sentence, i.e., either words or phrases; tree traversal algorithms find candidate nodes and features are extracted by analyzing the syntactic relationships conveyed by the parse tree. Not surprisingly therefore, parsing quality significantly impacts overall system quality, e.g., Toutanova et al. (2008) find that F-score increases from 78.2 to 88.4 when switching from automatic to gold parses.

2.3.2.0.2 Classifiers While Gildea and Jurafsky (2002) and others have developed special purpose classifiers for argument identification and classification, most state-of-the-art systems integrate standard discriminative classifiers such as the logistic classifier (Berger et al., 1996) or the support vector machine (Cortes and Vapnik, 1995), which are available 'off the shelf'. These classifiers have proven effective for the task and tend to outperform special-purpose generative models (Gildea and Jurafsky, 2002, e.g.,), also because it is straightforward to incorporate rich sets of features, such as those discussed in the next section.

2.3.2.0.3 Feature Engineering Feature engineering has been a central topic in frame-semantic analysis and state-of-the-art systems incorporate sophisticated sets of features. We can distinguish between *lexical* features, such as the argument head lemma, and *syntactic* features, which can further be divided into *clause-level* features that apply to the whole clause, such as the verb voice, and *argument-level* features that apply only to the particular argument, e.g., the argument part-of-speech. Due to its importance, the syntactic position of an argument is commonly encoded and incorporated into the classifier in several different ways, e.g., as the relation governing the argument node or as the full path of syntactic relations leading from the predicate node to the argument node, or as the linear position within the syntactic frame, etc. (see Swanson and Gordon, 2006). A list of features that have been used for both argument identification and classification is shown in Table 2.1. Note that the optimal set of features

Feature	Description
Verb	Verb (lemma) governing the argument.
Verb voice	Indicates active or passive voice.
Syntactic frame	The syntactic frame and the arguments position within this frame, e.g., np+vp+NP for a noun phrase appearing after the verb phrase.
Syntactic subcategorization	The phrase structure rule used to expand the parent of the predicate constituent.
Predicate-relative position	The surface position of the argument relative to the predicate constituent (left or right).
Distance to predicate	Some measure of the distance between the argument constituent and the predicate constituent.
Path from argument to predicate	The minimal path in the parse tree from the argument to the predicate node.
Path to common ancestor with predicate.	Especially the minimal path in the parse tree from the argument node to the lowest common ancestor with the predicate node.
Projected path	Path from maximum extended projection (the highest VP in the chain of VPs dominating the predicate) of the predicate to an argument.
Argument head	Head word (lemma) of the argument and its part-of-speech.
Argument lexical items	Non-head words of the argument and their part-of-speech.
Phrase type	The phrase type of the argument constituent.
Argument marker	Markers (especially the preposition) used for argument realization.
Additional lexical features	Features of relevant lexical items (verb head, argument head, etc.) obtained from semantic resources like WordNet, through a cooccurrence analysis, named entity recognition, etc.
Features of node relatives	Head word and part-of-speech, phrase type, etc. of left and right siblings as well as parent.
Further linking features	E.g., the part-of-speech of the subject, a cue which indicates missing subjects, and so on.

Table 2.1: Features used in argument identification and classification. Some of the features are specific to a constituent-based representation. Note that the optimal set of features for the two subtasks can differ and features should therefore be selected individually for each of the two subtasks.

for the two subtasks can differ and features should therefore be selected individually for each of the two subtasks. The list is compiled from Gildea and Jurafsky (2002), Xue and Palmer (2004), Toutanova et al. (2008) and Màrquez et al. (2008). Feature interactions are not listed, although they do provide additional information beyond the basic features and are thus important.

2.3.2.0.4 Global Optimization As a more recent development, major improvements have been achieved with models that find globally optimal role assignments (Toutanova et al., 2008), resulting from the understanding that a frame is a joint structure, with strong dependencies between the arguments. Global models, for example, enforce the constraint that each semantic role occurs at most once in a frame. One approach for implementing such constraints is by generating multiple global hypotheses (i.e., role assignments) with a purely local model, and scoring each hypothesis with a separate global model that gives preference to globally consistent hypotheses, a method known as *reranking* (Collins and Koo, 2005).

In a similar vein, global optimization also involves the joint optimization of all decisions along the processing pipeline, including those regarding syntactic analysis, which strongly influence system quality, as was pointed out in Section 2.3.2.0.1. Although integrating over parse trees would be the principled way of dealing with parser uncertainty, this is computationally not feasible. Therefore, it is also common here to work with multiple hypotheses (parse trees), each of which is forwarded along the pipeline to generate multiple possible outputs, the best of which is selected by a global model (Toutanova et al., 2008). Another possibility is to combine a multitude of models through an ensemble method such as boosting (Màrquez et al., 2005). These methods increase the robustness against parser errors, since syntactic and frame-semantic analysis are mutually informative and consequently parser uncertainty can be reduced via information from the frame-semantic component, and not just vice-versa. This idea can be taken one step further by developing models for joint syntactic and frame-semantic parsing, in which syntactic and semantic decisions interact more closely, by integrating both into a single component, rather than a pipeline. This currently constitutes a promising and active area of research (Merlo and Musillo, 2008; Titov et al., 2009; Xavier et al., 2009; Boxwell et al., 2010).

2.3.2.0.5 Relieving the Data Requirements A major factor that constricts the rapid development of frame-based language understanding systems is the time-intense knowledge engineering or data labeling effort, which has to be repeatedly invested for each new domain, genre and language. For instance, systems trained on PropBank demonstrate a marked decrease in performance (approximately by 10 percentual points) when tested on out-of-genre data (Pradhan et al., 2008). Consequently, various previous work has been devoted to alleviate the amount of human effort necessary in constructing these systems.

In early work, Riloff and Schmelzenbach (1998) aimed at reducing the engineering effort for a rule-based system by inducing frame extraction patterns from an unlabeled corpus. Candidate extraction patterns are generated from a corpus and then presented to a human judge who accepts valid patterns and labels their extraction slots with semantic roles. The system exploits the fact that, while it is typically difficult and time-consuming for a human to elicit valid extraction patterns based on prior knowledge, judging the validity of candidate patterns is easier and faster.

More recently, a few approaches have been undertaken to combine labeled and unlabeled data in order to either improve the coverage of existing resources or port resources from one language into another. A framework known as *annotation projection* has become popular for devising such *semi-supervised* methods. The idea is to project annotations from a labeled source sentence onto an unlabeled target sentence within the same language (Fürstenau and Lapata, 2009) or across different languages (Padó and Lapata, 2009; van der Plas et al., 2011). These methods crucially rely on computing alignments between sentences, or more precisely between predicate-argument structures within these sentences, based on syntactic and semantic cues.

In a similar vein, but outwith annotation projection, Gordon and Swanson (2007) propose to increase the coverage of PropBank to unseen verbs by finding syntactically similar (labeled) verbs and using their annotations as surrogate training data. Swier and Stevenson (2004) introduced a semantic role labeling system which induces role labels following a bootstrapping scheme where the set of labeled instances is iteratively expanded using a classifier trained on previously labeled instances. Their method starts with a dataset containing no role annotations at all, but crucially relies on VerbNet

(Kipper et al., 2000) for identifying the arguments of predicates and making initial role assignments. VerbNet is a manually constructed lexicon of verb classes each of which is explicitly associated with argument realization and semantic role specifications.

While theoretically attractive, these semi-supervised methods do not yet offer a complete solution to the data acquisition bottleneck. When applied to monolingual data the improvements compared to the (fully) supervised setting are relatively modest, e.g., Fürstenau and Lapata (2009) report an increase in F-score of under 1% on FrameNet data. Similarly, cross-lingual projection of annotations is accompanied by a significant loss of data quality, e.g., when projecting gold standard annotations on gold standard parses from English to German the projected annotations attain an F-score of around 81% (Padó and Lapata, 2009).

While the discussion here has focussed on sentence-level frame-semantic analysis, recent work has also addressed the induction of document-level frames without supervision (Chambers and Jurafsky, 2011, 2009, 2008), by combining the predicate-argument structures of multiple sentences. There are two key elements to their approach. One is the identification of frequently cooccurring events expressed through verbs or nouns, for example the verbs *Search*, *Arrest*, *Plead*, *Convict*, *Sentence*, which together realize the backbone of a document-level *Prosecution* frame. The second element is the identification of the argument entities of these events and their classification according to their semantic role, which is based on the event-relative syntactic positions a particular entity occurs in throughout the document.

2.3.3 Frame Semantics and Reasoning

Language understanding involves more than just computing a semantic representation of a given language input. After computing such a representation, a *reasoning module* must make inferences which are relevant to the particular application at hand. Frame semantics addresses language analysis as well as reasoning and strikes a balance between *expressiveness*, i.e., the range of semantic phenomena which it can capture, and *feasibility*. With respect to language analysis it does not have to cope with difficult

phenomena such as quantifier scoping or negation, which pose a barrier when deriving full first-order logical forms. With respect to reasoning, it exposes a set of concepts in the form of semantic roles, which are abstract enough to allow for a relatively concise set of hand-written inference rules and thus it avoids the daunting task of acquiring large amounts of detailed world knowledge and the associated computational problem of running inference. Thus from the practical perspective, the relevant difference to other approaches is not so much founded in the use of frame structures as opposed to predicate-logic formulae, but rather in this particular tradeoff between expressiveness and feasibility. In fact, it is straightforward to explicitly incorporate semantic roles within first-order predicate logic for example within a Neo-Davidsonian event representation (Parsons, 1994), by including predicates such as *Agent* or *Patient* etc. as is illustrated by the following example:

(2.20) Carl repaired the motor within a week.

(2.21) $\exists e\ Repair(e) \wedge Agent(e, Carl) \wedge Patient(e, motor) \wedge Duration(e, week)$

2.3.4 Applications

In the following paragraphs we will give two examples of how open-domain frame-semantic analysis has benefitted applications.

2.3.4.0.6 Information Extraction The goal of *information extraction* systems (e.g., Hobbs et al., 1997) is to extract frame instantiations of specific frames such as *Bombing* or *Company Merger and Acquisition*, which typically span multiple sentences (see also Section 2.1.5). The traditional approach has been to specify low-level extraction patterns which directly operate on a *syntactic* representation of the input text, i.e., the word- or chunk-sequence or possibly a parse tree. The extracts obtained by applying the individual patterns are then combined together into a document-level frame. A drawback of this approach which we have already mentioned in the introduction of this section are the high costs of developing a set of patterns with sufficient accuracy

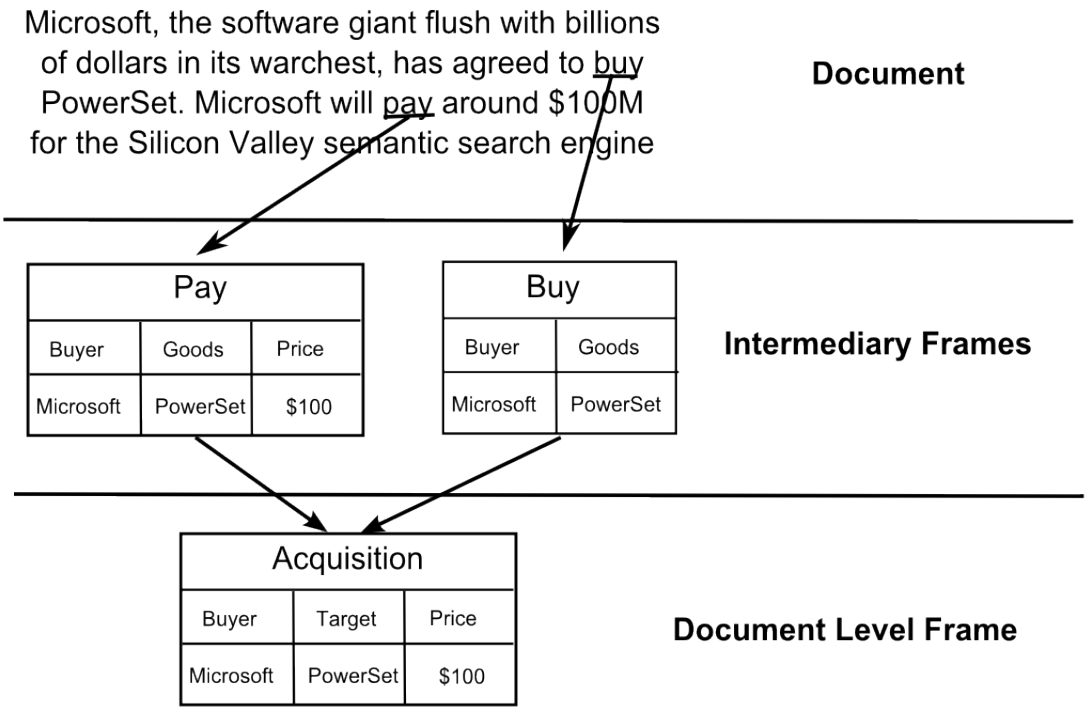


Figure 2.3: An illustration of how frame-semantic analysis can be used for information extraction. Surdeanu et al. (2003) propose a two-level architecture, in which input sentences are firstly mapped onto clause-level frames which are in turn mapped onto document-level frames. The clause-level frames serve as an intermediary representation which abstracts away from surface-level syntax and can be reused for any extraction task.

and coverage, which must be repeatedly created anew for each extraction task. Therefore, to alleviate portability, Surdeanu et al. (2003) propose a two-level architecture, in which input sentences are firstly mapped onto clause-level frames which are in turn mapped onto document-level frames. While the second mapping requires task-specific rules, computing the clause-level frames is domain-independent and can be achieved for example with a PropBank-trained model. Thus, the clause-level frames serve as an intermediary representation which abstracts away from surface-level syntax and can be reused for any extraction task. We have depicted this idea schematically in Figure 2.3.

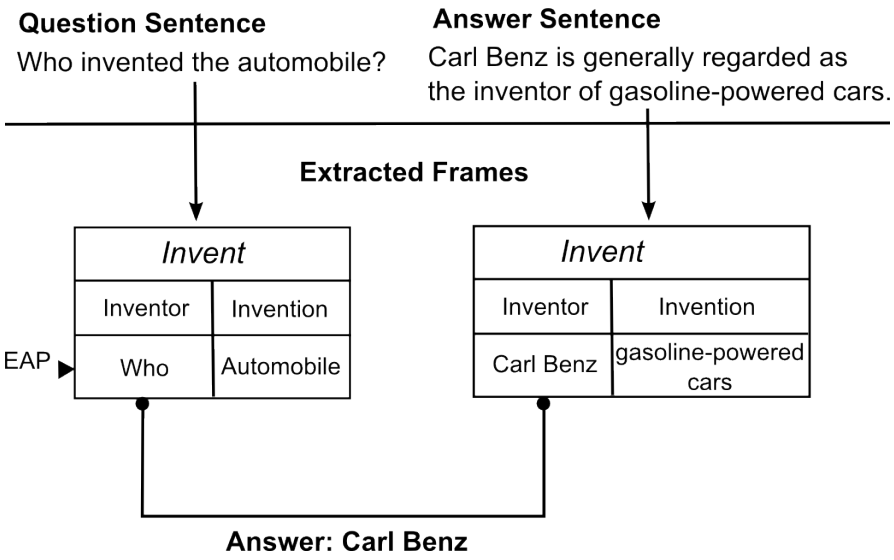


Figure 2.4: An illustration of how frame-semantic analysis can be used in question answering. Shen and Lapata (2007) propose an answer extraction method which firstly extracts frames from both question and answer sentences and then establishes a correspondence between them by aligning entities from the question and answer frame in order to find the answer phrase (*Carl Benz*) matching the *expected answer phrase* (*who*).

2.3.4.0.7 Question Answering A central problem for *question answering* systems is to bridge surface-level differences between a question such as *Who invented the automobile?* and an answer such as *Carl Benz is generally regarded as the inventor of gasoline-powered cars*. In addition to recognizing lexical paraphrases, e.g., *automobile* vs. *gasoline-powered cars*, a system must deal with syntactic variation which may lead to question-answer pairs with little syntactic resemblance. In contrast, we can expect correspondences at the semantic level between question and answer in terms of the frames and roles they express. Shen and Lapata (2007) follow this idea and propose an answer extraction method which firstly extracts frames from both question and answer sentences and then establishes a correspondence between them. Specifically, the method aligns entities from the question and answer frame (e.g., the *Invent* frame) according to their semantic role (e.g., *Cognizer*) in order to find the answer phrase (e.g., *Carl Benz*) matching the *expected answer phrase* (e.g., *who*), as shown in Figure 2.4.

In both of these examples clause-level frames serve as an intermediary representation which hides the complexity of syntax from downstream processing components. Further use cases of semantic roles include machine translation (Wu and Fung, 2009), coreference resolution (Ponzetto and Strube, 2006), summarization (Melli et al., 2005) and opinion expression detection (Johansson and Moschitti, 2010).

2.4 Summary

Frames represent situations by specifying participating entities and their *semantic roles*. As a linguistic theory, frame semantics describes how semantic roles are mapped onto the syntactic argument positions of a (verbal) predicate. Since this mapping aka *linking* from semantic roles onto syntactic positions can vary, a main concern is to account for the resulting variation in argument realization (*alternations*).

Frame-semantic analysis (aka semantic role labeling) is the task of automatically extracting frames from input sentences and labeling arguments with semantic roles. Current systems for frame-semantic analysis commonly employ data-driven models trained with supervised learning on large-scale resources such as PropBank or FrameNet. These resources, which are costly to construct, contain large amounts of hand-labeled sentences which document the possible mappings from semantic roles onto syntactic positions. Many systems rely on a cascade of classifiers, which identifies predicates and their arguments and labels them with their semantic roles. To this end both *syntactic* information and the *lexical* content of an argument are informative.

Both domain-specific and open-domain language understanding systems have been implemented on the basis of frame semantics which due to its shallowness is more practical than other approaches.

Chapter 3

Problem Setting

Before presenting models for *semantic role induction*, it is important to describe the main methodological issues which accompany the problem. Therefore in the following we will establish the setting in which our models are applied and evaluated.

We start by giving an exact definition of the *frame induction* problem and the subproblem of semantic role induction, which is the main concern of this thesis. Then we describe the datasets upon which we conduct our experiments for English, including the specific syntactic representation for input sentences, which in turn is closely tied to the predicate and argument identification tasks, which although not the focus of this thesis are discussed here since they are necessary for building an end-to-end system for frame induction.

Section 3.4 introduces an evaluation measure, called *collocation*, which together with the standard *purity* measure serves to assess the quality of induced semantic roles. Finally, Section 3.5 describes a baseline method for semantic role induction, which will serve as a point of comparison for the methods developed later in this thesis and provides an evaluation of that baseline. The dataset for German will be covered separately in Chapter 6.

3.1 Problem Formulation

Frame induction is the problem of computing a frame-semantic analysis without supervision in the form of annotations that indicate predicates, arguments, or argument roles and without relying on any other manually constructed semantic resources. In other words, the problem is unsupervised with respect to the frame-semantic analysis task. However, we assume that the input is syntactically analyzed in the form of a dependency tree, according to the syntax described below.

For the unsupervised setting we adopt the decomposition into three subproblems used in Sections 1.2 and 2.3.1 for the supervised setting. Predicate identification (Step 1) remains the same. Argument identification (Step 2) is now concerned with discarding non-semantic arguments, but does not make a final positive decision for any of the candidates. Therefore, while most candidates that pass this stage should be actual semantic arguments, some may also be non-semantic arguments. It is permitted that these instances are passed on to role induction, since there they can still be placed into a separate cluster for non-arguments.

Argument classification (Step 3) differs fundamentally from the supervised setting. Since in the unsupervised setting there is no predefined set of semantic roles, these must be induced from the data itself and we will refer to this problem as *role induction*.

Role induction follows the contract of a clustering problem in which the units selected by Step 2 are grouped into clusters representing semantic roles. The methods in this thesis will induce a separate set of clusters for each verb, i.e., the induced roles are verb-specific, much like the core roles in PropBank (see Section 2.2.2). After role induction a human could interpret and label each cluster. Alternatively, we can label clusters automatically by assigning identifiers such as R0, R1, etc. much like those used for PropBank core roles.

The output for a given input sentence consists of all extracted frame instantiations, each one specifying a verb and its arguments, including their role label which references a particular cluster of argument instances with the same semantic role.

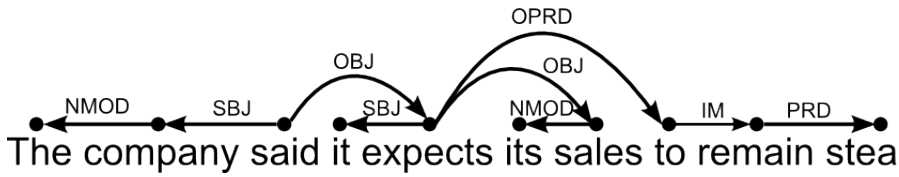


Figure 3.1: A sample dependency parse with dependency labels SBJ (subject), OBJ (object), NMOD (nominal modifier), OPRD (object predicative complement), PRD (predicative complement), and IM (infinitive marker).

3.2 Data

Evaluation for English is carried out on the gold role semantic annotations of the CoNLL 2008 (Surdeanu et al., 2008) training dataset. This dataset contains annotations for both verbal and nominal predicate-argument constructions, but we only evaluate against the former, as we are only concerned with verbal frame semantics.

The CoNLL dataset is taken from the Wall Street Journal portion of the Penn Treebank corpus (Marcus et al., 1993). Verbal frame semantic annotations are based on PropBank (Palmer et al., 2005), which is a natural choice of gold standard for our problem in which we aim to induce *verb-specific* roles.

The annotations have been converted from a constituent-based to a dependency-based representation (see Surdeanu et al., 2008). For each argument of a predicate only the head word is annotated with the corresponding semantic role, rather than the whole constituent. We will always take a content word to represent the head of the argument, rather than a function word (e.g. for prepositional phrases we take the nominal head rather than the preposition). We do not treat split arguments or coreferential arguments (commonly the case for relative clauses), i.e., we ignore arguments with a role that is preceded by the C- or R- prefixes used to indicate such arguments in the gold standard. Argument lemmas are normalized by converting to lower case, replacing numerical quantities with a placeholder and taking the most frequent lemma contained in a proper noun phrase as its head, in order to reduce data sparsity.

	auto/auto	gold/auto	auto/gold	gold/gold
Instances	240139	241557	224654	228129
Non-Arguments	49663	31382	0	0

Table 3.1: The number of instances and non-arguments in each of the four datasets, formed by combining automatic vs. gold parses with automatic vs. gold argument identification.

Input sentences are represented in the dependency syntax specified by the CoNLL 2008 shared task, which is illustrated through the example in Figure 3.1. A complete list of dependency labels together with a description can be found in Appendix C, Table C.1. The CoNLL 2008 dataset provides both gold and automatic parses, which we will use as alternatives in our experiments in order to assess the impact of parse quality on our methods.

In all our experiments we run a particular role induction method on the CoNLL 2008 training set and evaluate to what extent the induced clusters reflect the gold standard (details will follow in Section 3.4). We want to assess the performance on both gold vs. automatic parses and gold vs. automatic argument identification (discussed below) and therefore consider the four datasets corresponding the four possible combinations. Some basic statistics of these datasets are shown in Table 3.1.

3.3 Predicate and Argument Identification

While this thesis focuses on the *role induction* problem, unsupervised *frame induction* also comprises predicate identification and argument identification (see Section 3.1). Role induction is the most challenging of the three since it must take into account syntactic as well as lexical-semantic information, whereas predicate and argument identification can be viewed as purely syntactic processing steps that can be largely undertaken deterministically through a structural analysis of the dependency tree. Based on this understanding, this section develops a set of simple yet effective rules for identifying

1. Discard a candidate if it is a coordinating conjunction or punctuation.
2. Discard a candidate if the path of relations from predicate to candidate ends with coordination, subordination, etc. (see Appendix B for the full list of relations).
3. Keep a candidate if it is the closest subject (governed by the subject-relation) to the left of a predicate and the relations from predicate p to the governor g of the candidate are all upward-leading (directed as $g \rightarrow p$).
4. Discard a candidate if the path between the predicate and the candidate, excluding the last relation, contains a subject relation, adjectival modifier relation, etc. (see Appendix B for the full list of relations).
5. Discard a candidate if it is an auxiliary verb.
6. Keep a candidate if it is directly connected to the predicate.
7. Keep a candidate if the path from predicate to candidate leads along several verbal nodes (verb chain) and ends with arbitrary relation.
8. Discard all remaining candidates.

Table 3.2: Argument identification rules for English.

predicates and arguments for English.

Verbal predicates are relatively simple to identify based on their part-of-speech tags and thus the discussion in the following will concentrate on argument identification. As was described in Section 3.1, for the unsupervised setting we define argument identification task such that it is only concerned with filtering out as many non-semantic arguments as possible, but instances that pass this filter may still be labeled as bearing no role by the role induction component. Some supervised systems have adopted a similar definition (Koomen et al., 2005), although in most supervised systems the argument identification component makes a *final* positive or negative decision regarding the status of an argument candidate.

For English, we apply the rules given in Table 3.2 to discard or select argument candidates. They primarily take into account the parts of speech and the syntactic relations encountered when traversing the dependency tree from predicate to argument. A priori all words in the sentence are considered argument candidates for a given predicate. Then, for each candidate, the rules are inspected sequentially and the first matching rule is applied.

We will exemplify how the argument identification component works for the predicate *expect* in the sentence *"The company said it expects its sales to remain steady"* whose parse tree is shown in Figure 3.1. Initially, all words save the predicate itself are treated as argument candidates. Then, the rules from Table 3.2 are applied as follows. Firstly, the words *the* and *to* are discarded based on their part of speech (Rule 1); then, *remain* is discarded because the path ends with the relation IM and *said* is discarded as the path ends with an upward-leading OBJ relation (Rule 2). Rule 3 matches to *it*, which is therefore added as a candidate. Next, *steady* is discarded because there is a downward-leading OPRD relation along the path and the words *company* and *its* are discarded because of the OBJ relations along the path (Rule 4). Rule 5 does not apply but the word *sales* is kept as a likely argument (Rule 6). Finally, Rule 7 does not apply, because there are no candidates left.

On the CoNLL 2008 training set using gold parses these rules attain a precision of 87.0% and a recall of 92.1% whereas on automatic parses they attain a precision of 79.3% and a recall of 84.8%. Here precision measures the percentage of selected arguments which are actual semantic arguments and recall measures the percentage of actual arguments which are not filtered out. Note that these precision and recall scores are not exactly comparable to the ones reported in supervised systems, since a final decision about the argument status of these candidates has not been made (see above). In particular, the recall is relatively high compared to state-of-the-art supervised systems. For example, Màrquez et al. (2008) mention 81% recall, however for a constituent-based identification, which is more difficult. For a fair direct comparison we would have to take into account the results of role induction, during which some of these arguments may be assigned no role, thereby potentially increasing precision but also potentially decreasing recall.

Previous work by Grenager and Manning (2006) also devised rules for argument identification, but unfortunately, these are only mentioned and not documented in the paper. Recently, attempts have also been made to identify arguments without relying on a treebank-trained parser (Abend and Rappoport, 2010; Abend et al., 2009). Instead, they combine a part-of-speech tagger and an unsupervised parser in order to identify constituents and then determine likely arguments via a set of rules and by determining the degree of collocation with the predicate. Due to the fact that they do not rely on a treebank-trained parser, their method does not match the quality of a rule-based component which operates on parse trees produced by a supervised parser.

3.4 Evaluation

This section describes how we assess the quality of a role induction method, which assigns labels to the units which have been identified as likely arguments. As discussed in Section 3.1, each label simply indicates the cluster that the particular unit has been assigned to. Therefore, since the assigned labels do not have a prior interpretation, we cannot directly verify the correctness of each label by comparing to the gold standard label. Instead, we will look at the induced clusters as a whole and assess their quality in terms of how well they reflect the assumed gold standard. Specifically, for each verb, we determine the extent to which argument instances in the clusters share the same gold standard role (purity, see Manning et al., 2008) and the extent to which a particular gold standard role is assigned to a single cluster (collocation).

More formally, for each group of verb-specific clusters we measure the purity of the clusters as the percentage of instances belonging to the majority gold class in their respective cluster. Let N denote the total number of instances, G_j the set of instances belonging to the j-th gold class and C_i the set of instances belonging to the i-th cluster. Purity can be then written as:

$$PU = \frac{1}{N} \sum_i \max_j |G_j \cap C_i| \tag{3.1}$$

Collocation is the symmetric counterpart to purity and defined as follows. For each

gold role, we determine the cluster with the largest number of instances for that role (the role's *primary* cluster) and then compute the percentage of instances that belong to the primary cluster for each gold role:

$$CO = \frac{1}{N} \sum_j \max_i |G_j \cap C_i| \tag{3.2}$$

Per-verb scores are aggregated into an overall score by averaging over all verbs. We use the micro-average obtained by weighting the scores for individual verbs proportionately to the number of instances for that verb.

Finally, we use the harmonic mean of purity and collocation as a single measure of clustering quality:

$$F_1 = \frac{2 \cdot CO \cdot PU}{CO + PU} \tag{3.3}$$

Purity and collocation measure essentially the same data traits as precision and recall, which in the context of clustering are however defined on pairs of instances (see Manning et al., 2008). We find that this makes them a bit harder to grasp intuitively and therefore prefer purity and collocation. The same holds for other evaluation metrics, e.g. information-theoretic measures such as the V-Measure (Rosenberg and Hirschberg, 2007).

Purity and collocation should always be assessed in combination or together with F-score since one can be traded off against the other. Purity can be trivially maximized by mapping each instance into its own cluster while collocation can be trivially maximized by mapping all instances into a single cluster.

At the same time, while it is desirable to report model performance with a single score such as F-score it is equally important to assess how purity and collocation contribute to this score. In particular if the system were to be used for annotating data, low collocation would result in higher annotation effort while low purity would result in lower data quality. Therefore high purity is imperative for an *effective* system whereas high collocation contributes to *efficient* data labeling. For assessing our models we therefore introduce the following terminology. If a model attains higher purity than the

baseline, we will say that it is *adequate*, since the induced roles adequately represent semantic roles. If a model attains higher F-score than the baseline, we will say that it is *non-trivial*, since it strikes a tradeoff between collocation and purity that is non-trivial. Our goal then is to find models which are both adequate and non-trivial.

In addition to reporting overall aggregates, we will also (where appropriate) present results for 12 verbs which we selected so as to exhibit varied occurrence frequencies and alternation patterns: *say*, *make*, *go*, *increase*, *know*, *tell*, *consider*, *acquire*, *meet*, *send*, *open* and *break*.

We will also report per-role scores, whose interpretation requires some caution since core roles are defined individually for each verb and do not necessarily have a uniform corpus-wide interpretation. Thus, conflating per-role scores across verbs is only meaningful to the extent that these labels actually signify the same role (which is mostly true for A0 and A1). Furthermore, the purity scores we will provide in this context are averages over those clusters for which the specified role is the majority role.

3.5 Baseline Method for Semantic Role Induction

In Section 2.1.3 we discussed that the linking between semantic roles and syntactic positions is far from random. Consequently there is a strong tendency to map a particular semantic role into a specific syntactic position such as *Subject*, *Object* or into a *Prepositional Complement* using a particular preposition (Levin and Rappaport, 2005; Merlo and Stevenson, 2001). To further underline this statement we show in Table 3.3 how frequently individual semantic roles map onto certain syntactic positions, here simply defined as the relation governing the argument. The frequencies were obtained from the CoNLL 2008 dataset and are aggregates across predicates. As can be seen, there is a clear tendency for a semantic role to be mapped onto a single syntactic position. This is true across predicates and even more so for individual predicates. For example, A0 is commonly mapped onto *Subject* (SBJ), whereas A1 is often realized as *Object* (OBJ).

Algorithm 1: Baseline Method for Semantic Role Induction

input : argument instances for a particular verb

output: verb-specific clusters of instances

```
S ← the N most frequent syntactic positions in the dataset
foreach s ∈ S do
    allocate a cluster c_s for s
end
allocate a default cluster c_⊥ for all other positions
foreach instance x do
    s_x ← syntactic position of x
    if s_x ∈ S then
        assign instance to cluster c_{s_x}
    end
    else
        assign instance to default cluster c_⊥
    end
end
return all clusters
```

This motivates a baseline which directly assigns instances to clusters according to their syntactic position. The pseudo-code is given in Algorithm 1. For each verb we allocate $N = 22$ clusters (the maximal number of gold standard clusters plus a default cluster). Apart from the default cluster, each cluster is associated with one particular syntactic position and all instances occurring in that position are mapped into the cluster.

While the baseline is simple, the following chapters will show that it is quite difficult to outperform, confirming previous work which has reached the same conclusion (Grenager and Manning, 2006). This is largely due to the fact that almost $2/3$ of PropBank arguments are either $A0$ or $A1$ and thus by far most important distinction to make is between these two roles. Since this can to large extent be achieved on the basis of the arguments' syntactic position (as brought forward by Table 3.3), the baseline suc-

	SBJ	OBJ	ADV	TMP	PMOD	OPRD	LOC	DIR	Total
A0	50473	3350	145	4	2464	28	12	0	60398
A1	18090	50986	3207	45	4819	3489	118	170	83535
A2	1344	2741	6413	74	774	2440	606	800	19585
A3	88	254	1208	37	116	114	63	940	3359
A4	6	20	351	7	79	34	28	2089	2687
A5	0	0	19	0	1	3	0	28	67
AA	10	1	0	0	1	0	0	0	13
ADV	7	46	7364	33	55	31	103	2	8070
CAU	3	6	215	14	5	0	8	0	1178
DIR	0	3	304	2	5	1	19	639	1123
DIS	0	3	3326	47	2	0	15	0	4823
EXT	1	6	418	0	6	3	23	4	621
LOC	18	32	358	15	127	2	5076	9	5831
MNR	7	54	2285	22	59	36	154	6	6238
MOD	9	2130	77	22	69	3	6	0	9030
NEG	0	0	3078	39	0	0	0	0	3172
PNC	1	11	458	4	4	292	8	4	2231
PRD	0	2	41	0	0	11	2	0	66
PRT	0	0	0	0	0	0	0	0	2
REC	0	5	8	0	0	0	0	0	14
TMP	14	93	969	14465	141	1	42	15	16086
Total	70071	59744	30248	14830	8730	6488	6285	4706	228129

Table 3.3: Contingency table between syntactic position and semantic roles. Only the 8 most frequent syntactic positions are listed. Counts were obtained from the CoNLL 2008 training dataset using gold standard parses. The marginals in the rightmost column also include counts of unlisted co-occurrences.

cessfully reflects this aspect of the task and can achieve high scores, as the evaluation in the next section will show.

	Baseline		
	PU	CO	F1
auto/auto	68.3	72.1	70.1
gold/auto	74.9	78.5	76.6
auto/gold	77.0	71.5	74.1
gold/gold	81.6	78.1	79.8

Table 3.4: Baseline scores on the four datasets.

3.5.1 Baseline Evaluation

The overall scores for the baseline on English are shown in Table 3.4. As expected, gold parses result in higher scores than automatic parses. Supervised systems show similar improvements (see Section 2.3.2.0.1). Per-verb scores and per-role scores on the auto/auto dataset are shown in Table 3.5. These results confirm our assertion that due to the close correspondence between semantic roles and syntactic positions the baseline can attain relatively high scores.

3.6 Summary

We have defined the *frame induction problem* and the subproblem of *role induction*, which is the primary concern of this thesis and can be viewed as a clustering problem. Frame induction also comprises predicate and argument identification for which we have developed a rule-based component. We discussed the CoNLL 2008 dataset upon which we will test our models on English. The dataset contains annotations from PropBank, which is an appropriate choice of gold standard for our models which induce *verb-specific* semantic roles. We introduced an new evaluation measure, called *collocation*, which together with the standard *purity* measure serves to assess the quality of induced semantic role clusters. Finally, we described a baseline that identifies semantic roles with syntactic positions which despite of its simplicity attains high scores and

		Baseline		
Verb	Freq	PU	CO	F1
say	16698	86.7	90.8	88.7
make	4589	63.3	71.0	67.0
go	2331	47.3	56.0	51.3
increase	1425	58.0	69.0	63.0
know	1083	58.3	70.8	63.9
tell	969	59.0	76.8	66.7
consider	799	60.7	65.3	62.9
acquire	761	70.7	78.4	74.4
meet	616	70.0	72.2	71.1
send	515	68.3	67.4	67.9
open	528	55.3	67.8	60.9
break	274	51.1	59.1	54.8

(a) Per-verb scores.

		Baseline		
Role	Freq	PU	CO	F1
A0	49956	68.2	89.6	77.5
A1	72032	77.5	75.2	76.3
A2	16795	65.7	71.4	68.4
A3	2860	45.4	81.8	58.4
A4	2471	61.6	86.1	71.8
A5	44	46.4	59.1	52.0
AA	9	46.7	100.0	63.6
ADV	5824	33.8	86.3	48.6
CAU	878	67.5	79.3	72.9
DIR	811	51.5	71.6	59.9
DIS	3022	36.1	90.4	51.6
EXT	536	46.9	91.0	61.9
LOC	4481	65.1	76.5	70.4
MNR	5066	62.0	64.6	63.3
MOD	8064	80.2	44.1	56.9
NEG	2952	38.7	98.6	55.6
PNC	1682	67.9	71.8	69.8
PRD	56	39.1	92.9	55.1
REC	9	25.0	100.0	40.0
TMP	12928	71.1	78.7	74.7
NONE	49663	57.1	47.3	51.8

(b) Per-role scores.

Table 3.5: Fine-grained scores for the baseline on the auto/auto dataset.

is hard to outperform.

Chapter 4

Feature-based Probabilistic Models

Semantic role induction can be formulated as the problem of inferring the unobserved semantic role of an argument, given a set of informative features, e.g. the argument's syntactic position or its head word. By treating the features as well as the semantic role of an argument as random variables we can rely on probabilistic inference as a principled means of inferring an argument's semantic role. The challenge then consists of finding a set of valid assumptions regarding how the features and the semantic role of an argument relate to each other, which is the goal of this chapter.

We will develop two types of probabilistic models. In the first type, semantic roles are directly modeled as latent (unobserved) variables and related probabilistically to other clause-level and argument-level features. This approach benefits from the fact that role induction directly corresponds to probabilistic inference: the semantic role of an argument is determined by inferring the value of the latent variable. Since this type of model fully encapsulates the problem we are guaranteed to obtain good results, provided that we have found an adequate model. We will discuss our latent variable models in Section 4.1 and the related model of Grenager and Manning (2006) in the related work section at the end of this chapter.

The second type of probabilistic model is built around several linguistic assumptions regarding the empirical traits of how semantic roles and syntactic positions are linked

and adopts a layer of latent variables in order to generalize from the observed syntactic position of an argument to its semantic role by exploiting the close correspondence between the two. While this approach is less direct, it more clearly relates to linguistic theory through a set of explicit assumptions. This model is discussed in the second part of this chapter, Section 4.2.

4.1 Semantic Roles as Latent Variables

Semantic role induction can be conducted through probabilistic models in which a latent, i.e., unobserved, variable directly represents the semantic role of an argument. The basic idea is to model the statistical relationships that hold between various argument features, including the semantic role of the argument, which is incorporated as a latent variable. The values of that latent variable correspond to semantic roles which can thus be determined by the means of statistical inference.

The approach described here is inspired by the seminal work of Grenager and Manning (2006), who conceived a latent variable model for semantic role induction. Due to their rigorous mathematical foundations (probability theory) such models have become popular for various unsupervised language learning problems (a classical example is part-of-speech induction, Merialdo, 1994).

4.1.1 Models

We will formulate several *probabilistic graphical models* (see Bishop, 2006, for an introduction), focussing on models of individual arguments rather than models of whole frames. Modeling arguments individually is a logical first step, since any frame-level model also requires an adequate argument-level model.

After discussing which argument-level *features* are included, we will address how these features can be incorporated into our model (either as input or as output vari-

ables) and the issue of *directed vs. undirected* edges and then we will specify the models in detail.

4.1.1.0.8 Features While feature-rich models can potentially attain higher performance than feature-poor models, modeling the interactions between features can be difficult. We therefore chose to (initially) incorporate only the most informative features into our model, namely the verb lemma (**VLem**), argument head word lemma (**ALem**), syntactic position (**SPos**) and the function word (**FWord**), which indicates the particular lexical marker (preposition or infinitival *to*) with which the argument is realized.

Since alternation patterns are verb-specific and because we want to induce verb-specific semantic roles, the verb (lemma) must be included in the model.

As was pointed out in Section 2.1.4, the argument head word is often highly indicative of the underlying semantic role, in particular such words with a very specialized meaning (e.g. *sandwich* in the context of *eat*), whose occurrence with all but one particular role can be ruled out based on selectional constraints. Thus the argument head lemma is also incorporated into our models.

Similarly, the syntactic position correlates strongly with the semantic role and should equally be included in the model (see also Section 2.1.4). We will encode the particular syntactic position simply by the syntactic relation governing the argument word, i.e., one from the tables listed in Appendix C.

Finally, the function word involved in realizing an argument (in particular prepositions) is often understood as an marker for semantic roles and are therefore directly informative.

All of these features and the semantic role are naturally modeled as categorical variables, which assume one of several distinct possible values. For example, **ALem** ranges over all possible argument head word lemmas, and so on. The following paragraphs will discuss two possible ways in which these features can be incorporated: as

conditioning input variables or as generated output variables.

4.1.1.0.9 Input vs. Output features Our models are *discriminative* which means that certain features will be incorporated as (globally) conditioning variables, called *input variables*, in contrast to the *output variables*, whose values are generated by the model. Discriminative modeling can simplify dealing with complex interacting features, without making over-simplifying independence assumptions.

Moreover, the target likelihood function differs depending on whether a variable is included as input or output variable, and this in turn affects the induced latent values, i.e., semantic roles. Intuitively, the latent values of a model will be chosen such that the observed values are rendered maximally likely. Therefore, if a feature is incorporated as an output variable, the latent structure will adapt to the feature values in order to 'explain' those values. In contrast, when a feature is incorporated as an input variable, the latent structure (and the model as a whole) will not be guided to account for its values and only draws information from them in order to explain the output variables.

4.1.1.0.10 Directed vs. Undirected Edges Our models contain both directed and undirected edges. While directed edges are often used to encode *causal* relationships (or likewise), we found it difficult to relate the argument features in such a manner. As an example consider the relationship between the semantic role and the argument head. In accordance with linguistic theory we could assume that a particular role selects for its argument head and therefore model the relationship through a directed edge **Role** $\rightarrow$ **ALem**. While this is justified by the fact that a particular role can be viewed as imposing selectional constraints on its fillers, we find it counter-intuitive from a cognitive perspective that upon generating a sentence, the role is determined before the actual lexical content. In general, we found it difficult to give a stringent theoretical argument for or against particular edge directions.

Furthermore, equivalent models (specifying the same probability distribution) can be built with either directed or undirected edges. Nevertheless, we do employ directed edges in our model for efficiency reasons, since they involve locally normalized poten-

tials and thus help avoid computationally expensive global normalization terms (details follow below). In our models, directed edges however have no linguistic interpretation.

4.1.1.0.11 Model Structure We assume that, given the semantic role, the lexical content of an argument and its syntactic position are independent. On one hand, lexical selection is primarily a semantic phenomenon (see Section 2.1.3), i.e., it is the semantic role and not the syntactic position which constrains the set of possible argument heads. On the other hand, the distribution over possible linkings between semantic roles and syntactic positions is a property of the verb and does not depend on the specific lexical content of the arguments. All of our models incorporate this independence assumption.

A second important insight is that due to the close correspondence between semantic roles and syntactic position, it is reasonable to include the latter as an output variable. This will put 'pressure' on **Role** to become predictive of **SPos**. In fact, if **SPos** were the only output variable and directly connected to **Role**, the latter would be chosen to reflect the syntactic position as closely as possible, which is essentially the baseline solution.

To avoid this inherent limitation we need to include at least one other output variable, either **ALem** or **FWord** or both. Choosing only **FWord** as additional output variable, as for Model (a) in Figure 4.1, does not fundamentally change the pressure put onto **Role** and we cannot expect induced roles to differ much from the baseline.

This changes if we instead incorporate **ALem** as output variable, which results in Model (b). Here we can expect the induced semantic roles to differ more strongly from the baseline solution, due to the pressure resulting from **ALem**. Alternatively, we can incorporate *both* **ALem** and **FWord** as an output variables, which results in Model (c). The differences in performance between Model (b) and (c) are hard to anticipate a priori and will therefore be discussed together with the results.

For these Models (a)-(c), a role is assumed to select for the same argument head words across verbs. Similarly, the linking of roles onto positions is independent of the verb. Especially this latter assumption contradicts our linguistic understanding, that linking

preferences are verb-specific. Therefore, we conceived three further models with verb-specific linkings, which can be implemented as in Figure 4.1 (d)-(f) by connecting all the variables **Role**, **SPos** and **VLem**. Otherwise these models are analogous to Models (a)-(c) in how they include **ALem** and **FWord**.

4.1.1.0.12 Probabilistic Formulation A model defines a probability distribution over observed outputs Y and latent variables Z, conditional on the inputs X. This distribution can be written as a product of factors:

$$p(y,z|x) = \prod_i \varphi_i(x,y,z) \times \frac{1}{Q(x)} \prod_j \Psi_j(x,y,z) \quad . \tag{4.1}$$

The product contains two types of factors, φ_i and Ψ_j. The factors φ_i are (locally) normalized and directly express a conditional probability distribution between a variable V and its parents W:

$$\varphi_i(v,w) = p(v|w) \quad . \tag{4.2}$$

These distributions are implemented as multinomial (aka *categorical*) distributions, which leads to a set of multinomial parameters $\gamma_{v,w} = p(v|w)$. We will write γ_i for all the multinomial parameters of factor φ_i and γ for a parameter vector that comprises all γ_i.

The unnormalized factors Ψ_j express potentials between variables $V_1 \ldots V_N$ in the form of exponentials:

$$\Psi_j(v_1,\ldots,v_N) = \exp\left[\theta_j^\top \phi_j(v_1,\ldots,v_n)\right] \quad . \tag{4.3}$$

As all variables are discrete-valued, the sufficient statistics ϕ_j simply indicate the particular state that the variables $V_1 \ldots V_N$ are in, i.e., $\phi_j(v_1,\ldots,v_n)$ is a vector of indicator functions, one for each possible joint state. For each such indicator function $\phi_{j,k}$, indicating a particular state, there is a parameter $\theta_{j,k}$ which quantifies the (local) preference for that state. We will write θ for a parameter vector which comprises all θ_j.

The partition function $Q(x)$ normalizes the product over the unnormalized factors:

$$Q(x) = \sum_{y,z} \prod_j \Psi_j(x,y,z) \quad . \tag{4.4}$$

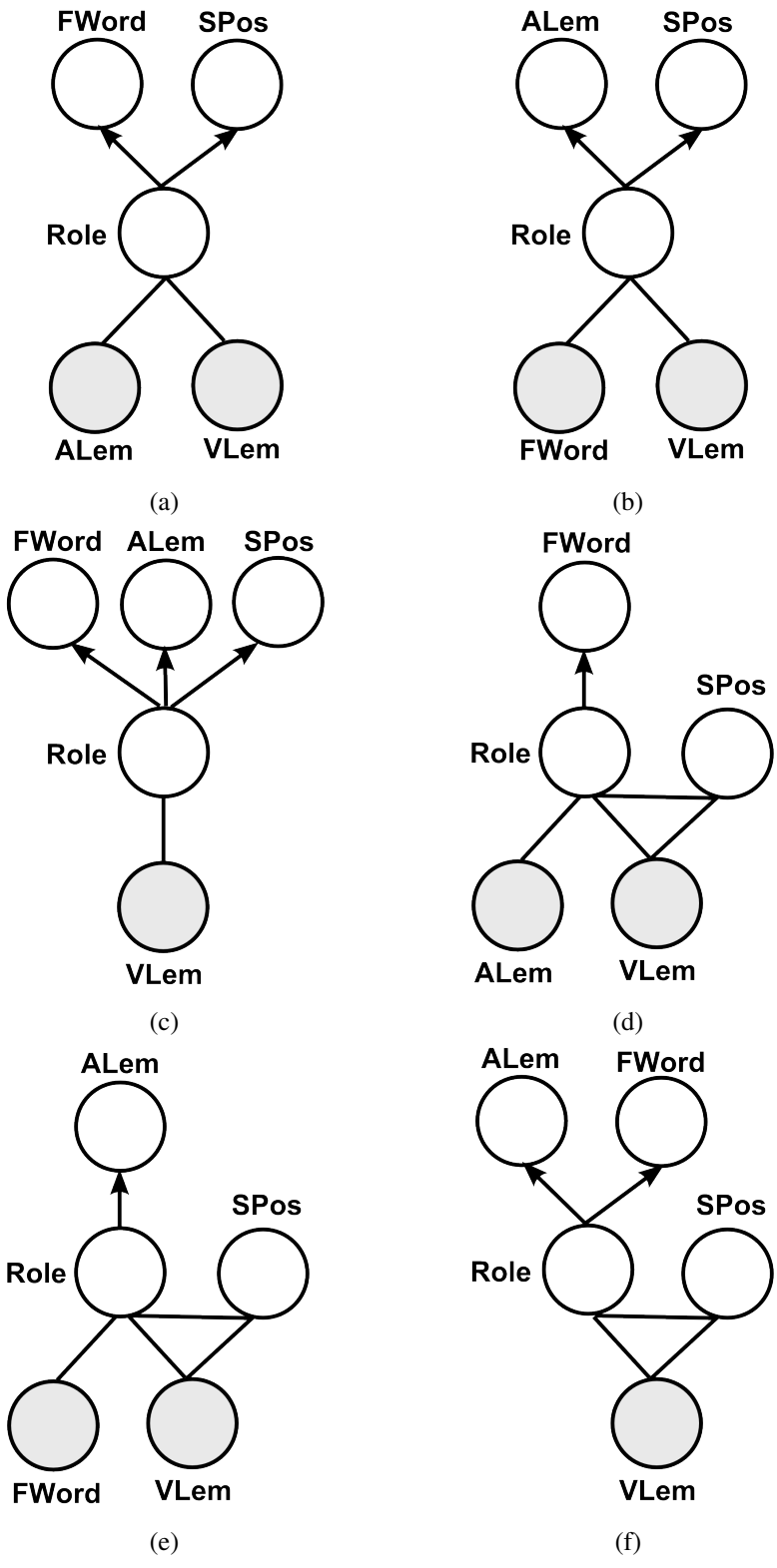


Figure 4.1: Six latent variable models for role induction. Input variables (i.e., conditioning variables) are drawn shaded and the variable **Role** is latent.

The directed edges in our models are advantageous over undirected edges in terms of efficiency, due to the fact that when computing the marginals, the locally normalized factors require no normalization and thus no summation over the values of the observed variables which participate in the factor. Especially when the range of a variable is large (e.g. for **ALem**) this results in significantly faster inference.

4.1.1.0.13 Parameter Estimation An optimal parametrization of our model can be found by determining the Ω^* which maximizes the log-likelihood of a training dataset of instances $(c^{(n)}, d^{(n)})$ consisting of inputs $c^{(n)}$ and observed outputs $d^{(n)}$:

$$\Omega^* = \arg\max_{\Omega} \sum_n \log \sum_z p(d^{(n)}, z | c^{(n)}, \Omega) \quad . \tag{4.5}$$

Here we have used $\Omega = (\gamma, \theta)$ for a parameter vector comprising all model parameters. Due to the hidden variables, we cannot solve this problem analytically but instead resort to Expectation-Maximization (EM, Bishop, 2006; Dempster et al., 1977), whereby the parameters are iteratively updated by maximizing the *expected* log-likelihood of the data:

$$\Omega^{(t+1)} = \arg\max_{\Omega} \sum_n \sum_z p(z | c^{(n)}, d^{(n)}, \Omega^{(t)}) \log p(d^{(n)}, z | c^{(n)}, \Omega) \quad . \tag{4.6}$$

In writing down this equation we have made use of the fact that instances are independent. The log probability occurring in this equation can be written out as:

$$\log p(y, z | x, \Omega) = \sum_i \log \varphi_i(x, y, z) + \sum_j \log \Psi_j(x, y, z) - \log Q(x) \quad . \tag{4.7}$$

In Equation 4.6, we must thus find the maximum over a (weighted) sum of summands. Now, since the parameters of the factors φ_i occur only in one summand (in particular they do not occur in the partition function), these summands can be maximized in isolation.

The optimization problem thus decomposes into isolated optimization problems for each of the factors φ_i:

$$\gamma_i^{(t+1)} = \arg\max_{\gamma_i} \sum_n \sum_z p(z | c^{(n)}, d^{(n)}, \Omega^{(t)}) \log \varphi_i(c^{(n)}, d^{(n)}, z) \quad , \tag{4.8}$$

and a joint optimization problem for the factors Ψ_j:

$$\theta^{(t+1)} = \arg\max_{\theta} \sum_n \sum_z p(z | c^{(n)}, d^{(n)}, \Omega^{(t)}) \left[\sum_j \log \Psi_j(c^{(n)}, d^{(n)}, z) - \log Q(c^{(n)}) \right] \quad . \tag{4.9}$$

Equation 4.8 can be solved analytically (i.e., we can do a full maximization step). For Equation 4.9 we need a numerical method. We experimented with both L-BFGS (Liu and Nocedal, 1989) and stochastic gradient ascent (Bottou, 2004) and chose the latter, since it yields just as good results but runs faster. The gradient $\nabla^{(n)}$ of each instance $(c^{(n)}, d^{(n)})$ is computed as

$$\nabla^{(n)} = E_Z[\phi] - E_{Y,Z}[\phi] \tag{4.10}$$

$$= \sum_z p(z|c^{(n)}, d^{(n)})\phi(c^{(n)}, d^{(n)}, z) - \sum_{y,z} p(z, y|c^{(n)})\phi(c^{(n)}, y, z) \quad . \tag{4.11}$$

These gradient contributions are scaled by a step size η, which is reduced after every EM iteration.

Conducting regularization upon each parameter update (i.e., for every instance) would be too inefficient, because it would require an update of all parameters, whereas the unregularized update only affects parameters with non-zero sufficient statistics. Therefore we conduct a batch-regularization, which after every M parameter updates scales parameters by a factor $0 < \xi = 1 - \eta M\lambda$. Here λ corresponds to the $L2$ regularization parameter which would be applied at each parameter update:

$$\theta^{(t+1)} = \theta^{(t)} + \eta(\nabla - \lambda\theta^{(t)}) \quad . \tag{4.12}$$

Our scheme thus approximates the total contribution of the regularization terms over M updates. Note also, that regularization decreases with the step size. This is consistent with the idea that regularization in our case helps avoid local optima in the search space, rather than improving generalization on separate test data. Due to the fact that parameter search is assumed to lead to better and better areas of the parameter space we can decrease regularization for the same reason that we decrease the step size.

4.1.1.0.14 Model and Parameter Settings Starting with randomly chosen initial parameters EM is run until convergence. The step size is adapted at each step by discounting the current value by a factor of $f = 0.95$, starting at an initial value of $\eta_0 = 1$. Regularization was carried out with parameters $\lambda = 0.00001$ and $M = 10000$.

4.1.2 Results and Analysis

The results of each model on the gold/gold dataset are shown in Table 4.1. Models (b) and (c) outperform the baseline in terms of purity and Models (d) and (e) outperform the baseline in terms of collocation. In terms of F-score Model (d) matches the baseline but purity is below the baseline whereas collocation is above. Furthermore, it is a priori clear that this model cannot induce significantly different roles to the baseline, because it only incorporates **FWord** as additional output, not however **ALem**. Therefore, none of the models is both adequate and non-trivial, i.e., none of the models simultaneously attains higher F-score and purity (see Section 3.4 for the definition of these terms).

While the verb-specific linkings implemented by Models (d)-(f) are theoretically more sound (see the discussion above), the joint factor between **VLem**, **Role** and **SPos** introduces a large number of parameters resulting in a model with much more degrees of freedom. This implies a greater potential for overfitting, in particular in the presence of data sparsity which is characteristic for our training setting. Model (e), which has to generate the two output variables **ALem** and **SPos**, exhibits the effects most drastically. Since the linking between **SPos** and **Role** can differ for each verb and can therefore be adapted more easily to the data, role induction will be strongly biased towards inducing roles that are predictive of **ALem**. In other words, the main cue for determining the semantic role of an argument is then its argument lemma, which results in low-purity clusters because the syntactic position is neglected.

Table 4.2 shows the scores on all datasets for the best-performing Model (d). While for gold argument identification the model matches or outperforms the baseline in terms of F-score, the model remains below the baseline on automatic argument identification. The non-arguments contained in these datasets therefore negatively affects performance. Across datasets purity is below the baseline whereas collocation is above the baseline.

Table 4.2 also shows the scores that the model obtains when it is trained on gold standard labels. Note that since the model is tested on the same dataset that it is trained on, the scores essentially measure the model's capacity to *memorize* the data, rather

	PU	CO	F1
Baseline	81.6	78.1	**79.8**
Model (a)	78.8	74.2	76.4
Model (b)	**82.4**	51.5	63.4
Model (c)	**82.4**	49.8	62.1
Model (d)	75.4	84.7	**79.8**
Model (e)	50.3	**89.8**	64.5
Model (f)	74.2	59.5	66.1

Table 4.1: The results of all models on the gold/gold dataset.

	Baseline			Model (d)			Memorize		
	PU	CO	F1	PU	CO	F1	PU	CO	F1
auto/auto	**68.3**	72.1	**70.1**	63.2	**77.9**	69.8	79.8	80.2	80.0
gold/auto	**74.9**	78.5	**76.6**	68.3	**83.4**	75.1	86.3	86.6	86.5
auto/gold	**77.0**	71.5	74.1	72.9	**80.5**	**76.5**	88.9	89.1	89.0
gold/gold	**81.6**	78.1	**79.8**	75.4	**84.7**	**79.8**	90.5	90.7	90.6

Table 4.2: Results for the best-performing (in terms of F-score) Model (d) on all datasets. As point of reference we also show scores of this model for supervised training ('Memorize'), which give an indication of the model's capacity at memorizing the data.

than its generalization properties. The scores reveal that the model is inherently limited in the sense that even under these training conditions it makes errors on around 10% of instances (gold/gold). This suggests that the incorporated features are not fully informative of the semantic role and a more precise model would require further features. For completeness, the per-verb scores and per-role scores for Model (d) on the auto/auto dataset are shown in Table 4.3.

To conclude, our attempts at directly modeling semantic roles as latent variables have been unsuccessful. Even for the simple feature sets and model structures we considered, it was difficult to build up a stringent argument in favor of one particular model and to anticipate and analyze the 'behavior' of a model. In addition, issues such as data

		Baseline			Model D		
Verb	Freq	PU	CO	F1	PU	CO	F1
say	16698	**86.7**	90.8	**88.7**	86.0	**91.6**	**88.7**
make	4589	**63.3**	71.0	67.0	62.6	**75.2**	**68.4**
go	2331	**47.3**	56.0	51.3	44.9	**64.5**	**52.9**
increase	1425	**58.0**	69.0	63.0	57.9	**71.5**	**64.0**
know	1083	**58.3**	70.8	**63.9**	53.1	**75.0**	62.2
tell	969	**59.0**	**76.8**	**66.7**	49.1	73.6	58.9
consider	799	**60.7**	65.3	**62.9**	51.2	**77.2**	61.6
acquire	761	**70.7**	78.4	**74.4**	63.6	**81.5**	71.4
meet	616	**70.0**	72.2	**71.1**	66.2	**76.0**	70.8
send	515	**68.3**	67.4	**67.9**	64.1	**71.7**	67.7
open	528	**55.3**	67.8	**60.9**	50.4	**70.6**	58.8
break	274	**51.1**	59.1	**54.8**	47.1	**64.6**	54.5

(a) Per-verb scores for Model (d).

		Baseline			Model D		
Role	Freq	PU	CO	F1	PU	CO	F1
A0	49956	**68.2**	89.6	77.5	66.0	**94.7**	**77.8**
A1	72032	**77.5**	75.2	**76.3**	67.3	**81.5**	73.7
A2	16795	**65.7**	71.4	**68.4**	60.2	**71.8**	65.5
A3	2860	45.4	81.8	58.4	**46.3**	**82.1**	**59.2**
A4	2471	61.6	**86.1**	**71.8**	**61.8**	85.3	71.7
A5	44	46.4	59.1	52.0	**59.0**	**84.1**	**69.4**
AA	9	46.7	**100.0**	63.6	**66.7**	**100.0**	**80.0**
ADV	5824	**33.8**	86.3	**48.6**	33.2	**86.4**	48.0
CAU	878	**67.5**	79.3	**72.9**	57.3	**80.3**	66.9
DIR	811	**51.5**	71.6	59.9	49.7	**79.0**	**61.1**
DIS	3022	36.1	90.4	51.6	**37.6**	**90.8**	**53.2**
EXT	536	46.9	**91.0**	61.9	**85.6**	90.3	**87.9**
LOC	4481	65.1	**76.5**	70.4	**67.2**	74.5	**70.7**
MNR	5066	62.0	**64.6**	63.3	**63.6**	63.3	**63.4**
MOD	8064	**80.2**	44.1	56.9	58.9	**91.7**	**71.7**
NEG	2952	38.7	98.6	**55.6**	**39.3**	**98.8**	56.2
PNC	1682	**67.9**	71.8	69.8	66.7	**74.0**	**70.2**
PRD	56	39.1	**92.9**	55.1	**40.0**	**92.9**	**55.9**
REC	9	**25.0**	**100.0**	**40.0**	0.0	**100.0**	0.0
TMP	12928	**71.1**	78.7	74.7	69.7	**82.7**	**75.7**
NONE	49663	**57.1**	47.3	51.8	52.4	**52.5**	**52.5**

(b) Per-role scores for Model (d).

Table 4.3: Fine-grained scores for the baseline and Model (d) on the auto/auto dataset.

sparsity and non-convexity of the optimization problem affect the practicability of this approach.

4.2 Semantic Roles as Canonical Syntactic Positions

For the probabilistic models in the previous section, role induction simply corresponds to inferring the values of the latent semantic role variable. In contrast, the approach described in this section is less direct and revolves around a linguistically motivated framework in which the key step of generalizing from an argument's observed syntactic position to its unobserved semantic role is implemented through a probabilistic latent structure model.

Roughly speaking, we postulate that arguments have a *canonical* syntactic position, onto which they are 'normally' mapped (e.g. *Agent* is normally mapped onto *Subject*). Triggered by special circumstances (e.g. *Passivization*), alternations may however lead to a deviation from this standard mapping. In such cases, the actual syntactic position of an argument differs from its canonical position and the goal then is to infer the argument's canonical position. Thereafter arguments can be grouped together according to their canonical position in order to obtain semantic role clusters. In the following, we will firstly describe the assumptions underlying our approach as well as the details of this conceptualization of the role induction problem and then present a model which can be applied in this setting.

4.2.1 Standard Linkings and Canonical Syntactic Positions

We build on the notion established in Section 2.1.4.1 that alternations are the result of differing underlying linkings. Recall that a linking is defined as the deterministic mapping from semantic roles onto syntactic positions. When two clauses employ a different linking, the same semantic role may be realized in different syntactic positions (see Figure 2.1 for an example). Despite alternations, we empirically observe a

strong tendency to map a particular semantic role into a particular syntactic position, as discussed previously when we defined our baseline in Section 3.5, which also makes use of this property. This can be explained by positing the existence of a *standard linking*, which is used distinctly more often than any other linking and which therefore gives rise to the high degree of correlation between syntactic positions and semantic roles. The syntactic position of an argument under the standard linking is called the argument's *canonical position*. We additionally assume that each possible semantic role can be realized under the standard linking.

Linkings, including the standard linking, are invertible, i.e., no two semantic roles are mapped onto the same syntactic position. Therefore, each canonical position can be understood as a 'proxy' for a particular semantic role. This allows us to formulate semantic role induction as a primarily syntactic process, which firstly determines the canonical position for each argument and then groups arguments by their canonical position. Our method thus attempts to transform clauses into a canonical syntactic form, from which it is trivial to label arguments with semantic roles. In the terminology of transformational grammar (Chomsky, 1965), our method can be understood as an attempt to reverse the transformations occurring in the transformational component.

Crucially, we need a model which implements the aforementioned idea and determines the canonical position for a given argument. Since canonical positions (just like semantic roles) are not observed we must exploit the fact that the standard linking occurs most frequently, and thus for most arguments the observed syntactic position is identical to the argument's canonical position. To this end we develop a model which generalizes from the argument's observed syntactic position to its canonical position.

4.2.2 Logistic Classifier with Latent Variables

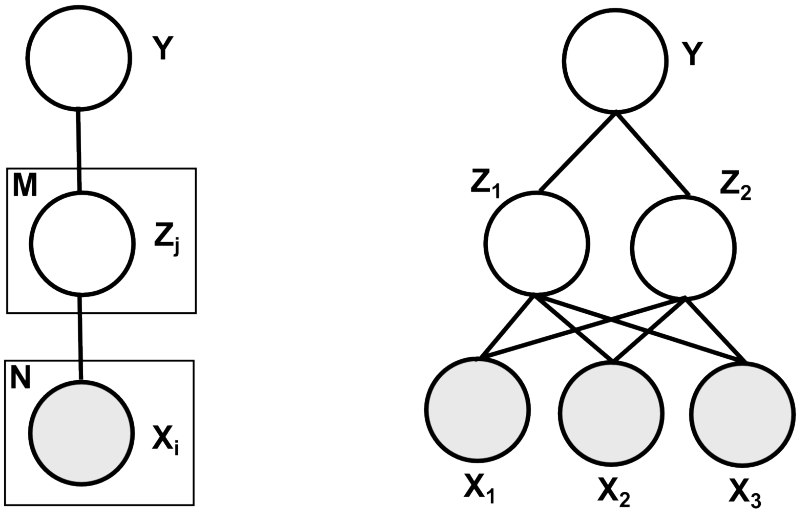
This section describes a probabilistic classifier which for a given set of argument features determines the canonical position of the argument. Importantly, our model implements a mechanism which allows it to generalize from the outputs seen during training, namely the argument's syntactic position, to the argument's canonical position. This is

achieved via a layer of latent variables, which is meant to capture the abstract argument properties which determine its canonical position.

An alternative formulation of the problem arises from the view that we are confronted with a supervised learning problem with noisy targets. While most instances are correctly labeled with their canonical position, this is not the case for arguments involved in alternations. Our model must be capable of adequately dealing with the noisy targets by correcting them to their canonical position. Admittedly, this is a very challenging machine learning problem and we are not aware of related work on a similar problem or have a full understanding of the fundamental limitations that may hold for such a setting.

Importantly, our model is informed only by local argument features, which are extracted at or below the node representing the argument head in the parse tree (apart from the verb lemma). This restriction guarantees that the features give no cues about possible alternations whose presence would allow the model to learn to produce output closer to the observed syntactic rather than canonical position of an argument. Consequently, the model has to rely primarily on the lexical content of an argument, which is however an informative source of information as was pointed out in Section 2.1.4.0.1. The specific features that are incorporated into the model are described in Section 4.2.2.0.17.

Standard classifiers, such as the logistic classifier or support vector machines are not applicable in this setting, as they assume noise-free targets. Since our model has to be trained and applied on the same dataset these models would simply 'memorize' the input-output mapping instead of generalizing in the desired way. Therefore we propose a model with improved generalization capabilities that extends the logistic classifier with a layer of latent variables which mediate between the input variables and the target variable (see Figure 4.2). As a result, inputs and target are no longer directly connected and the information conveyed by the features about the target must be transferred via the latent layer. The number of latent variables crucially determines the generalization properties of the model by determining the capacity of the channel between inputs and outputs. With too few latent variables too little information will be transferred via the latent variables, whereas with too many latent variables generalization will degrade,



(a) (b)

Figure 4.2: The logistic classifier with latent variables illustrated as a graphical model using (a) plate notation and (b) in unrolled form for $M = 2$ and $N = 3$.

since the model can also adapt to the noise. The next section will define the model in detail.

4.2.2.0.15 Probabilistic Formulation The model, depicted in Figure 4.2, defines a probability distribution over the target variable Y and the latent variables Z, conditional on the input variables X:

$$p(y,z|x) = \frac{1}{Q(x)} \exp\left(\sum_m \theta_m \phi_m(x,y,z)\right) \tag{4.13}$$

The normalizing partition function is given by

$$Q(x) = \sum_y \sum_z \exp\left(\sum_m \theta_m \phi_m(x,y,z)\right) \tag{4.14}$$

Each latent variable Z_j is binary and each input X_i is real-valued. Since all factors are exponential, the resulting model is log-linear. The parameter vector θ contains a parameter θ_m for each sufficient statistics ϕ_m, of which there are two types:

1. $\beta_m(X_i, Z_j) : \mathbb{R} \times \{0,1\} \to \mathbb{R}$, between an input X_i and a latent variable Z_j;

2. $\gamma_m(Y, Z_j) : \mathcal{Y} \times \{0,1\} \rightarrow \mathbb{R}$, between the target Y and a latent variable Z_j.

Each γ_m is an indicator for a particular combination of the state of the latent variable Z_j and the state of the target Y. Each β_m takes value x iff. the latent variable is in a particular state and zero otherwise.

4.2.2.0.16 Parameter Estimation Let (c, d) denote a training set of inputs and corresponding targets. We obtain a parametrization of our model by finding the θ^* maximizing the data log-likelihood, which is given by

$$\begin{aligned} l(\theta) &= \log p_(d|c,\theta) \\ &= \textstyle\sum_n \log \sum_z p(d^{(n)}, z|c^{(n),\theta}) \\ &= \textstyle\sum_n \log \frac{\sum_z \exp\left(\sum_m \theta_m \phi_m(c^{(n)}, d^{(n)}, z)\right)}{Q(c^{(n)}, \theta)} \end{aligned} \quad . \tag{4.15}$$

Here and in the following equations the index n references a particular instance. We conduct stochastic gradient ascent (Bottou, 2004) to solve this optimization problem, which requires computing the gradient of the target. The gradient component of a parameter associated with a sufficient statistic $\beta(X_i, Z_j)$ is given by

$$\sum_n \sum_{z_j} p(z_j|d^{(n)}, c^{(n)}) \beta(c_i^{(n)}, z_j) - \sum_n \sum_{z_j} p(z_j|c^{(n)}) \beta(c_i^{(n)}, z_j) \quad . \tag{4.16}$$

And the gradient component of a parameter associated with a sufficient statistics $\gamma(Y, Z_j)$ is

$$\sum_n \sum_{z_j} p(z_j|d^{(n)}, c^{(n)}) \gamma(d^{(n)}, z_j) - \sum_n \sum_{y, z_j} p(y, z_j|c^{(n)}) \gamma(y, z_j) \quad . \tag{4.17}$$

Computing the gradient requires computation of the marginals which can be performed efficiently using belief propagation (Yedidia et al., 2003). Note that due to the fact, that there are no edges between the latent variables, the inference graph is tree structured and therefore inference yields exact results.

4.2.2.0.17 Features and Target Encoding Apart from the verb lemma the feature representation of an argument comprises only local argument features, as was pointed out in Section 4.2.2. Specifically the set of features extracted from the dependency parses consists of the verb lemma, the argument lemma, the argument part-of-speech, the preposition involved in dependency between predicate and argument (if there is one), the lemma of left-most/right-most child of the argument, the part-of-speech of left-most/right-most child of argument, and a key formed by concatenating all syntactic relations to the argument's children. The syntactic position which is used as a target for training is encoded simply through the governor relation of the argument. Although more complex encodings could be chosen, we found this one most appropriate for comparison with the baseline, which also directly uses the governor relation.

For example, the features for the argument *sales* in the sample sentence given in Figure 3.1 are [expect, sales, NNS, its, its, PRP$, PRP$, NMOD]. Note that in this example, since the argument has only one child, left-most and right-most child coincide. The target for this instance (observed syntactic function) is OBJ.

4.2.2.0.18 Model and Parameter Settings The search procedure is parametrized in terms of the step size η, which is adapted at each step by discounting the current value by a factor of $f = 0.95$, starting at an initial value of $\eta_0 = 1$. We do not regularize the target function, because the latent variables already provide a mechanism to prevent overfitting. This was confirmed by experiments in which regularization did not improve results. The specific instantiation of the model used in our experiments has 5 latent variables. With 5 binary latent variables we can encode 32 different target values, which seems reasonable for our set of syntactic positions which comprises around 37 elements.

4.2.3 Results and Analysis

The results of the canonicalization model are shown in Table 4.4. The model remains consistently below the baseline showing that it is not successful at canonicalizing argu-

	Baseline			Canonicalization		
	PU	CO	F1	PU	CO	F1
auto/auto	**68.3**	**72.1**	**70.1**	62.1	70.4	66.0
gold/auto	**74.9**	**78.5**	**76.6**	67.0	76.5	71.4
auto/gold	**77.0**	**71.5**	**74.1**	71.1	68.9	70.0
gold/gold	**81.6**	**78.1**	**79.8**	73.4	73.5	73.4

Table 4.4: Results of Canonicalization on all datasets.

ments. The scores however also indicate that the model is not simply reproducing the baseline. On the auto/auto dataset for example, the model output differs from the observed syntactic position for approximately 23% of all instances. We found that many of these instances correspond to difficult cases, for example instances whose argument head lemma can occur both in *Subject* or *Object* position (e.g. *company* in the context of the *Acquire*). In fact, these instances often correspond to cases of alternations, i.e. deviations from the standard linking. This seems an interesting finding which we investigate further in the following section.

4.2.3.1 Detecting Alternations

The results of the previous section motivate examining the performance of our model on a simpler subtask, namely that of *detecting alternations*. Hereby, we will only assess the model's ability to detect arguments that are not in canonical position and will not assess its capabilities of assigning the correct canonical position.

This is straightforward to implement, since our model signals alternations by outputting a canonical position that differs from the observed syntactic position. Instances for which this is the case are then filtered out and not assigned to any cluster. The scores for the resulting clustering are shown in Table 4.5, together with a baseline for which an equal number of randomly selected instances have been removed in order to ensure a fair direct comparison. The baseline scores are stable across multiple runs in which a different set of random instances is removed. We can see that by identifying

		Baseline			Filtered		
	Instances	PU	CO	F1	PU	CO	F1
auto/auto	185157/240139	68.7	72.4	70.5	**70.3**	**80.7**	**75.2**
gold/auto	164399/211557	75.4	78.9	77.1	**76.1**	**89.9**	**82.5**
auto/gold	144180/224654	77.9	72.3	75.0	**83.1**	**86.4**	**84.8**
gold/gold	165327/228149	81.2	78.3	80.1	**83.9**	**86.4**	**85.2**

Table 4.5: Scores attained by clustering instances according to their syntactic position and removing alternations according to our model (Filtered) compared to a baseline for which instances are randomly removed.

alternations with our model and removing them both purity and collocation increase significantly and are consistently above the baseline. The same is observed across verbs and roles as shown in Table 4.6. Thus, although the model cannot determine the canonical position of an argument it can be successfully employed for detecting alternations.

We think that the linguistically motivated framework for role induction presented in this section provides an appealing way of framing the problem. In contrast to the latent variable models in Section 4.1 the model presented here makes explicit reference to linguistic theory and incorporates a set of (a priori) reasonable assumptions. Unfortunately, the machine learning problem that arises from our formulation, i.e., abstracting from observed to canonical positions, is challenging and our model does not successfully implement this step. Our analysis however shows that the model is in fact grasping a central aspect of the task, by detecting instances involved in alternations. This is an important subtask of role induction and could by itself be useful for example in the context of active learning (see Tong, 2001), in order to identify 'difficult' instances that require hand-labeling.

	Baseline			Filtered		
Verb	PU	CO	F1	PU	CO	F1
say	86.6	90.7	88.6	**90.2**	**94.7**	**92.4**
make	63.0	70.7	66.7	**67.3**	**80.3**	**73.2**
go	**46.9**	56.3	51.2	43.5	**74.4**	**54.9**
increase	**58.1**	69.4	63.3	53.3	**78.8**	**63.6**
know	58.9	69.7	63.9	**59.9**	**81.1**	**68.9**
tell	58.8	76.2	66.4	**61.3**	**79.6**	**69.3**
consider	60.8	64.1	62.4	**62.1**	**71.4**	**66.4**
acquire	70.3	79.9	74.8	**74.0**	**84.6**	**79.0**
meet	69.0	71.1	70.0	**71.3**	**77.1**	**74.1**
send	67.5	66.2	66.9	**70.0**	**79.7**	**74.5**
open	54.7	65.9	59.8	**55.0**	**71.8**	**62.3**
break	**52.0**	58.8	**55.2**	48.6	**62.5**	54.7

(a) Per-verb scores.

	Baseline			Filtered		
Role	PU	CO	F1	PU	CO	F1
A0	68.7	89.7	77.8	**70.8**	**93.8**	**80.7**
A1	77.7	75.4	76.6	**79.5**	**83.2**	**81.3**
A2	63.9	71.7	67.6	**65.7**	**80.3**	**72.3**
A3	45.5	82.5	58.7	**47.6**	**90.3**	**62.4**
A4	62.9	87.1	73.1	**66.6**	**91.0**	**76.9**
A5	45.2	67.7	54.2	**51.1**	**77.4**	**61.6**
AA	**53.8**	**100.0**	**70.0**	50.0	100.0	66.7
ADV	**34.3**	86.3	**49.1**	32.7	**92.2**	48.3
CAU	66.0	80.2	72.4	**75.5**	**82.6**	**78.9**
DIR	**49.6**	73.2	**59.1**	38.0	**87.7**	53.1
DIS	36.0	89.9	51.4	**39.7**	**93.7**	**55.8**
EXT	**46.9**	92.1	**62.2**	44.2	**98.7**	61.0
LOC	66.6	77.1	71.5	**67.1**	**87.5**	**75.9**
MNR	**65.6**	65.7	**65.7**	48.5	**83.0**	61.2
MOD	**80.1**	45.1	57.7	78.4	**62.4**	**69.5**
NEG	**39.6**	98.5	**56.5**	37.8	**99.0**	54.7
PNC	66.6	72.4	69.3	**73.3**	**83.4**	**78.0**
PRD	40.0	90.0	55.4	**44.1**	**93.6**	**59.9**
REC	0.0	**100.0**	0.0	**33.3**	**100.0**	**50.0**
TMP	70.8	79.2	74.7	**76.1**	**85.9**	**80.7**
NONE	59.5	47.6	52.9	**61.1**	**56.5**	**58.7**

(b) Per-role scores.

Table 4.6: Fine-grained scores on the auto/auto dataset attained by removing alternations (Filtered) compared to a baseline for which instances are randomly removed.

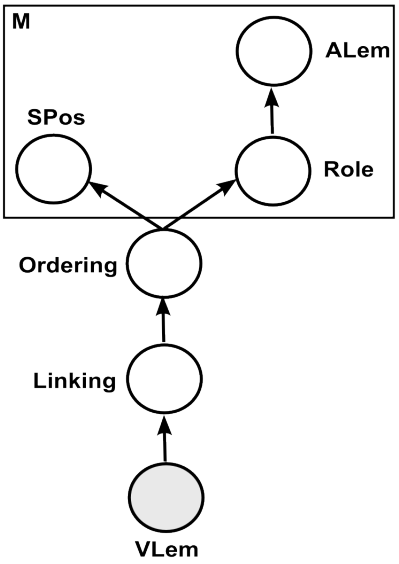


Figure 4.3: The model proposed by Grenager and Manning (2006) consisting of both an argument-level part and a frame-level part comprising the variables *Verb*, *Linking* and *Ordering*. The argument-level part comprises the variables *Role*, *SPos* (syntactic position) and *ALem* (argument lemma) and is replicated for each of the M arguments.

4.3 Related Work

Much like the models proposed in Section 4.1 the model of Grenager and Manning (2006), which is shown in Figure 4.3, incorporates latent variables which directly represent the semantic roles of arguments and can be used to induce verb-specific roles. Importantly, their model includes a frame level (super-)structure which combines all arguments occurring in a clause into a globally consistent frame. The argument-level part of their model is similar to that of Model (b) in Section 4.1 (see Figure 4.1b) and relates the semantic role, syntactic position and argument head lemma by assuming independence between the latter two conditional on the semantic role.

Frame-level information is represented in two variables: a linking variable, which captures the core roles and their mapping onto syntactic positions and an 'ordering' variable which additionally encodes the number and position of adjunct roles. The latter is generated from the linking variable in a process that inserts adjunct roles into the core

frame determined by the linking variable.

There are several tricky issues as to how exactly the conditional distributions in this model are defined. Firstly, linkings are generated by a construction process which for each of the five possible core roles A0-A4 samples and executes operations such as '*Add A0 to SBJ*', '*Add A1 to SBJ replacing AO*', '*Add A2 to Noun Phrase 1 shifting A1 to Noun Phrase 2*', and so on. The parameters which quantify the likelihood of each operation are chosen heuristically rather than adapted to the data and are shared across verbs, leading to a common prior over linkings for all verbs. Furthermore, the semantic role of an argument is fully determined by the ordering variable and likewise the syntactic position of core arguments is fully determined by the ordering variable. In other words, given the frame-level information the only uncertainty at the argument level is over lexical head words and over the syntactic position of adjuncts. Their model and evaluation furthermore only distinguishes between the different types of core roles and a single adjunct role, which subsumes all types of gold standard adjunct roles.

Grenager and Manning (2006) report improvements over a baseline that identifies syntactic positions with semantic roles (similar to the one described in Section 3.5), however with a type of evaluation that differs from the one used in this thesis. Most importantly, they do not measure collocation which makes it difficult to assess the overall performance of their model[1]. Looking at their purity scores, we see that they are once above the baseline[2] (on 'coarse roles', i.e., core roles and one adjunct role), and once below the baseline (on the core roles). Their findings are however consistent with ours from Section 4.1, where we showed that it is possible to conceive latent variable models which outperform the baseline in terms of either purity or collocation (but not both of them simultaneously).

Earlier work by Gildea (2002) developed several probabilistic models of argument

[1]Note that the *Recall* scores they report are not equivalent to collocation, as in the case of the 'Classification Only' task on the 'Coarse Roles' they are identical to the *Precision* (purity, in our terms) scores. This suggests that their recall is defined as $precision \cdot \alpha$ where α is the fraction of instances included in the set of arguments to evaluate. For the 'Classification Only' task on 'Coarse Roles' α is therefore trivially 1, since all arguments have either a core or adjunct role. In contrast, for 'Core Roles' some arguments with gold standard core role may have been omitted and α is less than one.

[2]We think their baseline scores on 'coarse roles' would be higher, if the baseline were not designed to rigidly assign all but a few syntactic positions onto *Adjunct*, as this clearly leads to low purity for this role.

structure which incorporate the verb lemma, argument head word and syntactic position as features. One of these models furthermore incorporates a latent variable that represents the semantic role of the argument and an additional latent variable which captures an abstract and not further defined class that the argument instance belongs to. The model is specifically designed to capture subject-object alternations, however similarly to our conclusions from Section 4.1 they conclude that "while models trained using the Expectation Maximization algorithm do well at fitting the data, the results may not correspond to the human analyses they were intended to learn."(p. 6).

Recently, Klementiev and Titov (2011) proposed a Bayesian model for unsupervised semantic parsing, which aims at learning frame-semantic representations, in contrast to previous work which was directed at learning lambda-calculus expressions for given inputs sentences (e.g. Zettlemoyer and Collins (2005) or Poon and Domingos (2009), i.a.). Their model jointly conducts argument identification and classification and in addition to predicting predicate-argument relationships, it also assigns each argument to a semantic class. These classes constitute an important part of their whole-frame, generative model, in which each predicate firstly generates the semantic class of an argument, which in turn selects a lexical realization. This differs from the models discussed in Section 4.1 in which semantic role and lexical realization are directly connected. Unfortunately, they do not directly assess the model's suitability for unsupervised frame-semantic parsing, but only indirectly evaluate the quality of the induced representations on a domain-specific question answering task, thereby leaving open to what extent the model is actually suited for inducing semantic roles.

4.4 Summary

We presented two feature-based probabilistic latent structure models. In the first model the semantic role of an argument is directly incorporated as a latent variable, whose value can be inferred by the means of probabilistic inference. We considered various different model structures but none led to induced clusters that are better than the baseline, showing that in order to successfully apply this approach more complex models are necessary.

In the second approach the goal is to determine the canonical syntactic position of an argument, which uniquely references a specific semantic role. Although we showed that by detecting alternations the model grasps a central aspect of the role induction task, the model does not determine the canonical positions of arguments correctly. Due to these difficulties with probabilistic feature-based models the next chapter will take a fundamentally different approach.

Chapter 5

Role Induction via Similarity-Graph Partitioning

The previous chapter revealed several shortcomings of the feature-based, latent structure approach applied to role induction. We encountered the difficulty of expressing our linguistic knowledge in terms of probabilistic relationships that hold between the involved features and consequently could not construct a well-performing model. This chapter describes a fundamentally different approach to role induction, that relies on judgements regarding the similarity of argument instances with respect to their semantic roles. Rather than modeling the relationship between argument features, we model when two argument instances have the same role or have differing roles. We argue that it is comparatively easy to formulate such similarity judgements and show that models based on them consistently outperform the baseline both in terms of F-score and purity.

In our 'similarity-driven' models all information about individual instances is encoded in similarity values to other instances and therefore it is not possible to represent instances in isolation, in contrast to the feature-based representation assumed in the previous chapter. A natural representation for such inherently relational data is a graph, whose vertices correspond to argument instances and whose edge weights express similarities. Based on this representation, we can formulate role induction as a graph partitioning problem, in which the goal is to partition the graph into clusters of vertices

representing semantic roles. Like in the previous chapter, we will induce verb-specific roles and therefore construct and partition a separate graph for each verb.

Our graph partitioning algorithms are based on two mechanisms that exploit the similarity information encoded in the graph. The first mechanism is *agglomeration*, in which two clusters containing similar instances are grouped together into a larger cluster. The second mechanism is *propagation*, in which role-label information is transferred from one cluster to another, based on their similarity. If we assume that the label for some cluster is known, then we can transfer that label (with some confidence) to other similar clusters, or conversely, we can inform dissimilar clusters that their label is likely to differ.

The chapter is organized as follows. Section 5.1 discusses how similarity is measured with similarity functions on argument instance pairs. Section 5.2.1 makes use of these similarity functions for defining the graph representation of our data. In Sections 5.3 and 5.4 we describe two algorithms for partitioning the graph into clusters representing semantic roles. The results of these algorithms are described and analyzed in Section 5.5. In Section 5.6 we describe and analyze an alternative graph partitioning approach that deviates from the algorithms described in Section 5.2 in that it relies on instance-wise similarities only, rather than cluster-wise similarities. Related work and a summary will follow in Section 5.7 and 5.8 respectively.

5.1 Measuring Similarity

The models in this chapter rely on judgements about the similarity or dissimilarity of the semantic roles of pairs of argument instances. Consider, for example, the two sentences below.

(5.1) Jim ate [a sandwich].

(5.2) [The sandwich] was eaten.

Evidently, the marked arguments have the same role, which can be inferred from their lexical content by virtue of the fact that *sandwich* is role-unambiguous in the context of the given verb *eat*. The reasoning here is the same as in Section 2.1.4.0.1 where we formulated the *one role per context* assumption, which states that for a particular predicate a given content word is commonly associated with a single semantic role. Generally, if arguments of the same predicate agree lexically, their semantic roles are likely to be the same.

As a second example consider the following two arguments occurring in the same sentence.

(5.3) Jim broke [the window] [with a hammer].

Here, we can assert that roles differ based only on the simple criterion that arguments occurring within the same clause are likely not to bear the same role.

Similarity judgements can also be based on the arguments' parts-of-speech, although less reliably. Like for the frame-criterion in Example (2), differing parts-of-speech provide *negative evidence*, i.e., indicate that the roles are not the same. In contrast, *positive evidence* is provided where arguments occur in the same syntactic position. These four types of similarity judgements based on the arguments' head words, parts-of-speech, syntactic positions and frame constraints will inform the models developed in this chapter. The following section will formalize the notion of similarity.

5.1.1 Similarity Functions

The similarities for a particular feature f (head word, part-of-speech, etc.) are measured with a similarity function $\phi_f(v_i, v_j)$, which assigns a value in $[-1, 1]$ to any pair of instances (v_i, v_j). Similarities are measured on an interval scale, i.e., while sums, differences and averages of the values of some similarity function ϕ_f express meaningful quantities, products and ratios do not. Moreover, the values of two distinct functions ϕ_{f_1} and ϕ_{f_2} cannot be meaningfully compared without rescaling.

Positive similarity values indicate that the semantic roles are likely to be the same, negative values indicate that roles are likely to differ and zero values indicate that there is no evidence for either case. The magnitude of ϕ_f expresses the degree of confidence regarding the similarity judgement and the extreme values -1 and 1 consequently indicate maximal confidence for the respective case.

Each similarity function ϕ_f can also be viewed as a (simple) classifier, which takes as inputs the feature values v_i^f and v_j^f of the two instances v_i and v_j on feature f and outputs a confidence-weighted decision, where the sign $sgn(\phi_f(v_i, v_j))$ indicates the decision (positive/negative) and the absolute value $|\phi_f(v_i, v_j)|$ quantifies confidence.

We could learn the similarity functions from a training dataset, but that would require (at least a small amount of) labeled data. Fortunately, the classifiers are simple enough that they can be specified directly based on prior knowledge, as is illustrated by the examples above. Specifically, we can use indicator functions which output either 1 or -1 iff. feature values are equal and 0 otherwise. For example, lexical similarity can be measured as

$$\phi_{lex}(v_i, v_j) = \begin{cases} 1 & \text{if } v_i^{lex} = v_j^{lex}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

Similarly, we can use an indicator function that outputs -1 iff. the arguments occur within the same frame:

$$\phi_{frame}(v_i, v_j) = \begin{cases} -1 & \text{if } v_i^{frame} = v_j^{frame}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

and similarly for the other two features. Despite of their simplicity, we will show that these four similarity functions are surprisingly effective at informing role induction.

5.2 Graph Partitioning

The similarity-driven models in this chapter formalize role induction as a graph partitioning problem, in which a graph whose vertices represent argument instances is

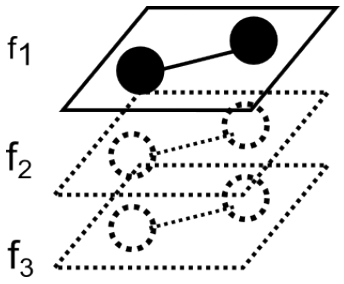


Figure 5.1: A multi-layer graph consists of multiple edge layers, one for each feature.

partitioned into vertex-clusters that represent semantic roles. The partitioning algorithm groups similar instances into the same cluster and dissimilar instances into to different clusters. It is informed by similarity information which comes from the similarity functions defined in the previous section and which is encoded into the graph as edge weights. The following two sections will firstly specify the details of our graph representation and the graph partitioning problem. On the basis of this graph representation, we will then formulate two role induction algorithms, which employ different partitioning mechanisms. Both algorithms determine the number of clusters automatically.

5.2.1 Graph Construction

Given the similarity functions for various features and a set of argument instances for a particular verb, we can construct a graph, whose vertices correspond to instances and whose edges represent similarity-relationships between the instances. Since each feature has its own similarity function, it is also associated with its own set of edges, and thus the graph consists of several layers of edges, one for each feature. This is illustrated schematically in Figure 5.1. The layer for a particular feature connects instance-pairs with non-zero similarity for that feature with an edge, whose weight quantifies the similarity between the instances with respect to the feature.

Assume there are M features, each associated with a given feature similarity function ϕ_f. A multi-layer graph is defined as a pair $(V, \{E_1, \ldots, E_M\})$ consisting of vertices

Figure 5.2: A schematic depiction of how the overall score $s(c_i, c_j)$ between two clusters is computed for agglomerative partitioning. In a first layer-wise aggregation step the edge weights between the two clusters (similarities between individual instances) are aggregated into a single score at each feature layer. The score provides the aggregated evidence in favor or against a merge collected for a particular feature. In a second step, the scores for all features are combined into a single overall score.

V and edge layers E_f. The set of vertices $V = \{v_1, \ldots, v_N\}$ consists of all N argument instances for a particular verb. The edge layer E_f for feature f is constructed by connecting all vertex-pairs with non-zero similarity with respect to f:

$$E_f = \{(v_i, v_j) \in V \times V | \phi_f(v_i, v_j) \neq 0\}. \tag{5.6}$$

Each edge $(v_i, v_j) \in E_f$ in layer f is weighted as $\phi_f(v_i, v_j)$, i.e., with the similarity value between the two connected vertices.

5.2.2 Problem Formulation

The graph partitioning problem, consists of finding a set of clusters $\{c_1, \ldots, c_S\}$ which form a partition of the vertex-set, i.e., $\cup_i c_i = V$ and $c_i \cap c_j = \emptyset$ for all $i \neq j$, such that (ideally) each cluster contains argument instances of only one particular semantic role, and the instances for a particular role are all assigned to one and the same cluster. The following sections will provide two solutions to the graph partitioning problem, that differ in terms of the basic operations they employ to partition the graph.

5.3 Agglomerative Graph Partitioning

This section describes a partitioning algorithm, which iteratively merges vertex clusters in order to arrive at increasingly accurate representations of semantic roles, following the general outline of a standard agglomerative clustering algorithm (see Jain et al., 1999). After initialization (discussed in Section 5.3.0.0.2), the algorithm starts with a clustering that has high purity but low collocation, i.e., in which the instances of a particular semantic role are scattered amongst many clusters. Then, collocation is iteratively improved by executing a series of merge steps, in which pairs of clusters are merged together. This requires a scoring function that quantifies how likely two clusters are to contain arguments of the same role. A key question is how to define this scoring function on the basis of the underlying graph representation, i.e., with reference to the instance similarities expressed by the edges. To this end, we take into account the connectivity of a cluster pair at each feature layer of the graph, in order to collect evidence for or against a merge. This crucially involves an aggregation over all edges which connect the two clusters, which allows inferring a cluster-level similarity score from the individual instance-level similarities encoded in the edges. The evidence collected at each layer must then be combined together in order to arrive at an overall decision (see Figure 5.2). The following sections will present the details of the algorithm and the scoring function.

5.3.0.0.1 Cluster Agglomeration Algorithm Essentially, the cluster agglomeration given in Algorithm 2 iteratively merges pairs of clusters until a termination criterion is met. The decision which cluster pair to merge at each step is made by scoring a set of candidate cluster pairs and choosing the highest scoring pair (Line 5). While it would be possible to enumerate and score all possible cluster pairs at each step, we apply a more efficient and effective procedure, in which the set of candidates consists of pairs formed by combining a fixed cluster c_i with all clusters c'_j larger than c_i. This requires comparing only $O(|C|)$ rather than $O(|C|^2)$ scores and more importantly it favors merges between large clusters whose score is more reliably computable, as will be described in the next section. Roughly speaking, the scoring function implements an averaging procedure over the instances contained in the clusters, which yields more

Algorithm 2: Cluster merging procedure. Operation $merge(c_i, c_j)$ merges cluster c_i into cluster c_j and removes c_i from the list C.

```
1  while not done do
2      C ← a list of all clusters sorted by number of instances in descending order
3      i ← 1
4      while i < length(C) do
5          j ← arg max_{0≤j'<i} s(c_i, c_j')
6          if s(c_i, c_j) > 0 then
7              merge(c_i, c_j)
8          end
9          else
10             i ← i + 1
11         end
12     end
13     update-thresholds
14 end
```

reliable (less noisy) scores when clusters are large, i.e., contain many instances. This prioritization therefore promotes reliable merges over less reliable merges in the earlier phases of the algorithm which in turn has a positive effect on merges in the later phases. Secondly, by keeping c_i fixed we relax the requirements for our scoring function, since we only require that $s(c_i, x)$ and $s(c_i, z)$ are comparable (i.e., where one cluster is argument in both scores), not however scores $s(w, x)$ and $s(y, z)$ between arbitrary cluster pairs. In the following two sections we will discuss initialization and the scoring function, two critical elements, which as will become clear are tied together for our problem.

5.3.0.0.2 Clustering Initialization A standard initialization for agglomerative clustering is to place each instance into its own cluster, resulting in an initial clustering with maximal purity and minimal collocation. There are two reasons which motivate

a more sophisticated initialization for our problem. Firstly, the scoring function we use is more reliable for larger clusters (this will be discussed in the following Section 5.3.0.0.3) than for smaller clusters. In fact, the standard initialization which creates clusters containing only a single instance each would not yield useful results as our scoring function crucially relies on initial clusters containing several instances on average. Secondly, as was described in Section 5.1, the similarities for differing features are not directly comparable and thus conceiving a scoring function which integrates different types of similarities poses a major challenge.

In our case, the four types of similarities based on the arguments' head words ϕ_{lex}, parts-of-speech ϕ_{pos}, syntactic positions ϕ_{syn} and frame constraints ϕ_{frame} are not as such comparable, and we have no means of composing them into a single score without resorting to heuristic judgements on how to weight each one. In particular, it is difficult to weight the contribution of the two forms of positive evidence given by lexical and syntactic similarity.

This brings forward the idea of using syntactic similarity for initialization, and lexical similarity for scoring. This separation avoids the difficulty of defining the exact interaction between the two. Specifically, we obtain an initial clustering by grouping together all instances which occur in the same fine-grained syntactic position, i.e., all pairs (v_i, v_j) for which $\phi_{syn}(v_i, v_j) = 1$.

Linguistically this is justified by the analysis in Section 2.1.4.1, which showed that the arguments occurring in a specific syntactic position under a specific linking share the same role. In other words, if we choose a set of syntactic cues which encode both the syntactic position of an argument and the employed linking we can assume that the arguments occurring in a particular fine-grained position encoded by these cues all bear the same semantic role. We adopt this analysis and assume that each of our fine-grained syntactic positions (roughly) corresponds to a specific position within a linking and define them as four-tuples consisting of the following cues:

- verb voice (active/passive);

- argument linear position relative to predicate (left/right);

- syntactic relation of argument to its governor;

- preposition used for argument realization.

Two positions are equal iff. they agree on all cues. While the incorporation of additional cues (e.g., indicating the part of speech of the subject or transitivity) would increase the initial purity, it would also create problematically small clusters, thereby negatively affecting the successive merge phase. Our specific choice here therefore is the result of a tradeoff between linguistic accuracy and practical applicability of the algorithm on our dataset. Note though that this is not a fundamental limitation, since applying our algorithm to larger datasets would relieve data sparsity by increasing the number of instances per cluster and therefore allow incorporation of further syntactic cues.

5.3.0.0.3 Cluster-Pair Scoring While the similarity functions defined in Section 5.1.1 measure role-semantic similarity *between instances*, the scoring function measures role-semantic similarity *between clusters*. Clearly, the similarity between two clusters can be defined in terms of the similarities of the instances contained in the clusters. This involves two aggregation stages: a first stage over the instance similarities in each feature layer, resulting in an aggregate score for each feature and then a second stage that integrates these scores into a single score, which quantifies the overall similarity between the two clusters (see Figure 5.2).

5.3.0.0.4 Layer-wise Aggregation Given two clusters, we can determine their similarity with respect to a particular feature f by analyzing the connectivity of the two clusters on the corresponding feature layer. Specifically we can average over the weights of edges between the two clusters. A common choice in graph clustering (Schaeffer, 2007) is to take the edge density between the clusters as a measure of cluster similarity by computing the average similarity between all pairs of instances in the clusters:

$$s_f(c_k, c_l) = \frac{1}{N_k \cdot N_l} \left(\sum_{v_i \in c_k} \sum_{v_j \in c_l} \phi_f(v_i, v_j) \right) \tag{5.7}$$

Here, N_k and N_l denote the number of instances in cluster c_k and c_l respectively. However, edge density is an inappropriate measure of similarity in our situation, since we cannot assume that arbitrary pairs of instances are similar with respect to a particular feature, even if the two clusters represent the same semantic role. Consider for example lexical similarity: most head words will not agree (even within a cluster) and therefore averaging between all pairs would yield low scores, regardless of whether or not the clusters represent the same role. Analogously for the dissimilarity based on frame constraints, the vast majority of instance pairs from the two clusters will belong to different frames, even if each instance in one cluster belongs to the same clause as an instance in the other cluster. Again, averaging over all possible pairs of instances would not yield indicative scores.

This motivates an averaging procedure in which for each instance in one cluster we find a maximally similar or dissimilar instance in the other cluster and average over the scores of these alignments:

$$s_f(c_k, c_l) = \frac{1}{N_k + N_l} \left(\sum_{v_i \in c_k} absmax_{v_j \in c_l} \phi_f(v_i, v_j) + \sum_{v_j \in c_l} absmax_{v_i \in c_k} \phi_f(v_i, v_j) \right) \quad (5.8)$$

Here *absmax* is a functional that returns the extremal value of its argument, either positive or negative: $absmax_{x \in X} g(x) = g(\arg\max_{x \in X} |g(x)|)$. Note that the alignments are unconstrained in the sense that $v_a \in c_k$ can be aligned to $v_b \in c_l$ in term 1 of Equation 5.8, while v_b can be aligned to some other instance in term 2. Moreover alignments in each term are many-to-one, i.e., multiple instances from c_k can be aligned to the same $v_b \in c_l$ in term 1 and similarly for term 2. This last point implies that score aggregation does not reflect the distributional properties of clusters, e.g. the occurrence frequencies of head words in each cluster. Consider for example two clusters containing an identical set of head words. Since many-to-one alignments are allowed each instance can be aligned with maximal score to some other instance regardless of the frequencies of these words.

However, it seems reasonable to assume that a particular semantic role imposes a specific distribution on the feature values of its instances, at least for features such as the argument head word, even though, for sparse features like the argument head word and the dataset sizes under consideration here, this assumption is not likely to result in significantly better scores in practice, because reliable frequency estimates are only

possible for large sample sizes. Nevertheless, for *lex* and *pos* we will also use cosine similarity as an alternative similarity measure between clusters:

$$s_f(c_k, c_l) = \frac{x_k^f \cdot x_l^f}{\|x_k^f\| \|x_l^f\|} \tag{5.9}$$

Here x_k^f and x_l^f are vector representations of the cluster containing as components the occurrence frequencies of a particular value of the feature f. Alternatively, we could enforce a one-to-one alignment constraint and redefine Equation 5.8 as the optimal bipartite matching between the two clusters. Apart from the fact that this would adhere to the graph formulation (in contrast to Equation 5.9) we see no theoretical argument that would justify its superiority over cosine similarity and moreover its computation would require cubic runtime in the number of vertices using the Hungarian algorithm (Munkres, 1957), which is prohibitively slow for sufficiently large clusters.

5.3.0.0.5 Layer Score Combination After computing the score for each layer according to the previous section, these scores need to be combined into an overall cluster similarity score. Due to the fact that the similarity scores and their aggregates for different features are not directly comparable (see Section 5.1.1) combining these scores through summation would require weighting each layer score according to its relative strength.

Unfortunately, the required weights are difficult to specify based on prior knowledge and therefore we propose an alternative scheme which is based on the distinction between positive and negative evidence introduced in Section 5.1. Negative evidence is used to strictly rule out a merge, whereas positive evidence provided by the lexical score is used as a graded measure to score merges which have not been ruled out:

$$s(c_k, c_l) = \begin{cases} -1 & \text{if } s_{frame}(c_k, c_l) < \alpha, \\ -1 & \text{if } s_{pos}(c_k, c_l) < \beta, \\ s_{lex}(c_k, c_l) & \text{if } s_{lex}(c_k, c_l) > \gamma, \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

When the part-of-speech similarity is below a certain threshold β or when clause-level constraints are satisfied to a lesser extent than threshold α, the score takes value -1 and the merge is ruled out. If the merge is not ruled out the lexical similarity score determines the magnitude of the overall score, provided that it is above the threshold γ. Otherwise, the function returns 0 indicating that neither strong positive nor negative evidence is available.

Much like the instance-similarity functions discussed in Section 5.1.1, the scoring function discussed in this section can also be viewed as a binary classifier, which outputs a decision regarding whether or not to merge a particular pair of clusters. The classifier is informed by the similarity scores for each feature layer and outputs a confidence-weighted decision (positive/negative), where the sign $sgn(\phi_f(v_i, v_j))$ indicates the decision and the absolute value $|\phi_f(v_i, v_j)|$ quantifies confidence. The scoring function given in Equation 5.10 implements a simple decision list classifier, whose decision rules are sequentially inspected from top to bottom, applying the first matching rule. Although this definition avoids weighting, it has introduced the threshold parameters α, β and γ, whose update we discuss in the next section.

5.3.0.0.6 Threshold Update Due to the lack of labeled training data we have no means of estimating the thresholds α, β and γ which parametrize the scoring function from data. We therefore determine a scheme in which the parameters β and γ are iteratively adjusted whereas the threshold α, which determines the extent to which the frame constraints can be violated, is kept fixed. Specifically we heuristically set $\alpha \leftarrow -0.05$, based on the intuition that in principle frame constraints must be satisfied although in practice, due to noise we have to expect a small number of violations (at most 5% of instances can violate the constraint).

The parameters β and γ are initially set to their maximal value 1, thereby ruling out all merges except those with maximal confidence. The parameters are then iteratively lowered according to a routine whose pseudo-code is specified in Algorithm 3. The parameter β is lowered at each iteration by a small value (0.025) until it reaches a value $\varepsilon = 0.025$, at which point its value is reset to 1.0 and the value of γ is discounted by a factor close to one (0.9). This is repeated until γ falls below ε upon which the algorithm

Algorithm 3: Update routine for the threshold parameters called after every iteration by Algorithm 2

```
1 β ← β − 0.025
2 if β ≤ 0.0 then
3 |   β ← 1.0
4 |   γ ← 0.9γ
5 |   if γ < ε then
6 |   |   done ← true
7 |   end
8 end
```

terminates.

5.3.1 Runtime Analysis

As was described in the previous section, Algorithm 2 stops when the threshold γ falls below some small value ε. Both γ and α are iteratively lowered based on a fixed schedule and therefore there is a constant value T for the number of steps of the outer loop that starts in Line 1.

Each pass through inner loop that starts at Line 4 iterates over $O(|C|)$ clusters and for each one a score with $O(|C|)$ other clusters is computed. The following argument shows that in total this requires $O(|V|^2)$ computations. Assume that f_i denotes the fraction of all V instances in cluster c_i, i.e., $f_i V = |c_i|$ and $\sum_{i=1}^{|C|} f_i = 1$. Then, overall

the number of instance-wise similarities we need to evaluate is at most

$$\begin{aligned}
&\sum_{i=1}^{|C|}\sum_{j=i+1}^{|C|}(f_i|V|)(f_j|V|)\\
=&\frac{1}{2}\sum_{i=1}^{|C|}\sum_{j=1}^{|C|}(f_i|V|)(f_j|V|)-\frac{1}{2}\sum_{i=1}^{|C|}(f_i|V|)^2\\
\leq&|V|^2\sum_{i=1}^{|C|}\sum_{j=1}^{|C|}f_i f_j\\
=&|V|^2\sum_{i=1}^{|C|}f_i\sum_{j=1}^{|C|}f_j\\
=&|V|^2 \qquad .
\end{aligned}$$

The total runtime in terms of the input quantities is therefore $O(T \cdot |V|^2)$. Although this could be prohibitively inefficient for large datasets, long runtimes were not a major concern in our experiments. As an optimization, the cluster similarity scores in Line 5 of Algorithm 2 can be cached such that they only need to be recomputed when a cluster changes, i.e., it is merged with another cluster.

5.4 Graph Partitioning by Label Propagation

This section describes an alternative partitioning algorithm which rather than greedily merging clusters is based on propagating cluster membership information amongst a set of initial clusters. There are two major advantages this algorithm has over agglomerative partitioning. On one hand it is less prone to make false greedy decisions which cannot later be revoked, as is the case for the merges of the agglomerative algorithm. While in general the 'locally optimal' decisions made by the agglomerative algorithm are correct, there are of course situations where it produces false results, in particular when scores are unreliable, i.e., for small clusters. More importantly however, the algorithm proposed has significantly lower runtime than the agglomerative algorithm which becomes important when inducing roles on larger datasets.

The algorithm is informed by the same similarity functions as the one in the previous

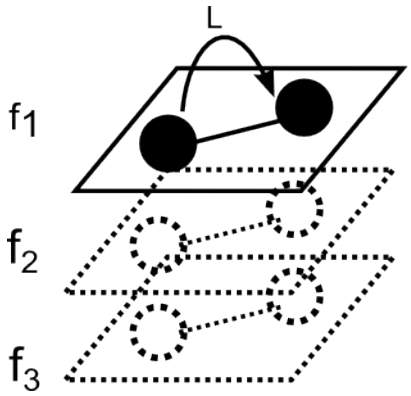


Figure 5.3: A schematic depiction of label propagation, in which role-label information is propagated between the vertices of a *propagation graph*, which comprise several vertices of the original data graph. Labels are propagated according to the similarity information contained on the different feature layers.

section but provides an alternative means of *cluster inference*. It is based on the idea of propagating cluster membership information along the edges of a graph, which is derived from the original multi-layer graph that represents the data. Each vertex of this derived graph, called the *propagation graph*, is assigned a label, that indicates which cluster the vertex currently belongs to. The propagation algorithm then proceeds by iteratively updating the label for each vertex, based on the labels of neighboring vertices and reflecting their similarity to the vertex being updated. The final labeling that results from running multiple iterations of node updates represents a partitioning of the graph into vertex-clusters of similar instances. In the next section we will firstly describe how the propagation graph is constructed from the original multi-layer data graph and then in Section 5.4.0.0.8 provide the outline of our label propagation algorithm, followed by the details of how vertices are updated. In Section 5.4.2 we will then relate the algorithm to the agglomerative algorithm described in the previous section.

5.4.0.0.7 Propagation Graph Construction A propagation graph is derived from the original data graph by collapsing several vertices of the original data graph into a single vertex of the propagation graph. Thus each vertex of the propagation graph represents an *atomic* set of instances of the original graph, that are always assigned to the same cluster. For our particular problem, the vertices of the propagation graph

correspond the clusters of vertices in the original graph that are obtained by grouping together instances by syntactic position, i.e., they are identical to the initial clusters of the agglomerative algorithm discussed in Section 5.3.0.0.2. Formally, let $a_i \in A$ denote the i-th vertex of the propagation graph, which references an atomic cluster of vertices $\{v_{i_1} \dots v_{i_{N_i}}\}$ of the original graph which occur in the same fine-grained syntactic position. Since each vertex of the propagation graph corresponds to a cluster of vertices in the original graph, the edges of the propagation graph can then be defined in terms of the edges between these vertices in the original graph. We can directly reuse Equations 5.8 and 5.9 to define the edge weights of propagation graph edges as aggregates over the edge weights in the original data graph. For each feature layer we define the set of edges as

$$B_f = \{(a_i, a_j) \in A \times A | s_f(a_i, a_j) \neq 0\}. \tag{5.11}$$

Each edge $(a_i, a_j) \in B_f$ in layer f is accordingly weighted as $s_f(a_i, a_j)$. In the following each vertex a_i will be associated with a label l_i indicating the partition that a_i (and consequently all the vertices in the original graph that have been collapsed into a_i) belongs to.

5.4.0.0.8 Label Propagation Algorithm Initially, each vertex of the propagation graph belongs to its own cluster i.e., we let the number of clusters $L = |A|$ and set $l_i \leftarrow i$. Given this initial vertex labeling, the algorithm proceeds by iteratively updating the label for each vertex (Lines 4-10 of Algorithm 4). This crucially relies on a scoring procedure in which a score $s(l)$ is computed for each possible label l (Line 5). The details of the scoring procedure will be described in the next Section. Intuitively, neighboring vertices vote for the cluster they are currently assigned to, where the strength of the vote is determined by the similarity (i.e., edge weight) to the vertex being updated. The vertex is assigned the highest scoring label, provided that its score is positive (Lines 6-8).

5.4.0.0.9 Label Scoring The label scoring procedure required in Line 5 of Algorithm 4 has parallels to the scoring procedure of the agglomerative algorithm for cluster pairs discussed in Section 5.3.0.0.3 and also consists of two stages: a first stage in which evidence is collected independently on each feature layer by computing label

Algorithm 4: Label Propagation Algorithm.

```
while not done do
    A ← a list of all propagation graph vertices sorted by size (number of contained
    instances) in descending order
    i ← 1
    while i < length(A) do
        l* ← arg max_{l∈{0...L}} s(l)
        if s(l*) > 0 then
            l_i ← l*
        end
        i ← i + 1
    end
    adjust thresholds
end
```

score aggregates with respect to each feature and a second stage in which these feature scores are combined in order to arrive at an overall score. We will discuss these two stages in the following.

5.4.0.0.10 Layer-wise Aggregation Assume we are updating vertex a_i. Then the first step is to compute the score for each feature f and each label l:

$$s_f(l) = \sum_{a_j \in \mathcal{N}_i(l)} s_f(a_i, a_j) \quad , \tag{5.12}$$

where $\mathcal{N}_i(l) = \{a_j | (a_i, a_j) \in B_f \wedge l = l_j \wedge |a_j| > |a_i|\}$ denotes the set of a_i's neighbors with label l, that are larger than a_i. Intuitively, each neighboring vertex votes for the cluster it is currently assigned to, where the strength of the vote is determined by the similarity to the vertex being updated. The votes of all (larger) neighboring vertices are counted together resulting in a score for each possible label. The condition of including only larger vertices for computing the score is analogous to the prioritization mechanism of the agglomerative algorithm described in Section 5.3.0.0.1, where

for a given candidate cluster only merges with larger clusters are considered. The argumentation is also similar, namely that scores for larger clusters are more reliable, although here there is also an opposing effect, since we are excluding neighboring vertices which might also provide valid evidence for or against a label. Nevertheless, in our experiments size-prioritization indeed contributed towards better scores, like in the agglomerative algorithm.

5.4.0.0.11 Layer Score Combination Given the scores $s_f(l)$ for a particular label l on each layer f the goal is to combine these scores into a single overall score $s(l)$ for the label. Like in Section 5.3.0.0.3 combining these scores through summation is not possible without 'guessing' weights and therefore we use a sequential combination instead:

$$s(l) = \begin{cases} -1 & \text{if } s_{frame}(l) < \alpha, \\ -1 & \text{if } s_{pos}(l) < \beta, \\ s_{lex}(l) & \text{if } s_{lex}(l) > \gamma, \\ 0 & \text{otherwise.} \end{cases} \tag{5.13}$$

Analogously to Equation 5.13, negative evidence that stems from the parts of speech or frame constraints can veto a propagation, whereas positive evidence stemming from the argument head words can promote a propagation. If neither strong negative nor positive evidence is available the label is assigned a score of zero. Note that scoring function is parametrized in terms of three parameters with an identical interpretation as those for the scoring function of the agglomerative algorithm. The threshold update that takes place in Line 11 of Algorithm 4 can therefore also be kept identical to the one described in Section 5.3.0.0.6 for the agglomerative algorithm.

5.4.1 Runtime Analysis

Let T denote the number of iterations of the outer loop which starts at Line 1 of Algorithm 4. The inner loop starting at Line 4 iterates over $|A|$ clusters and for each has to evaluate at most $|A|$ neighboring nodes. Additionally, there are the one-time costs of computing the similarities between the atomic clusters which, following the same ar-

gument as in Section 5.3.1 costs $O(|V|^2)$. The total costs are therefore $O(T|A|^2 + |V|^2)$. Since $|A|^2 << |V|^2$ the runtimes of label propagation are significantly lower than those of agglomerative clustering.

5.4.2 Comparison with Agglomerative Clustering

Our description of the label propagation algorithm already made explicit reference to the agglomerative algorithm discussed in Section 2. Both algorithms are informed through identical similarity functions and use analogous aggregation and scoring procedures. A key difference is that the merge operations of the agglomerative clustering algorithm are irreversible, whereas labels are reassigned at each iteration to atomic clusters in the label propagation algorithm. While the asymptotic runtime of both algorithms is the same ($O(TV^2)$), label propagation runs faster in practice since it does not require recomputing cluster similarity scores for a merged cluster pair with all other clusters but instead only requires a one-time computation of all scores between atomic clusters.

5.5 Results and Analysis

The results of both the agglomerative and the label propagation algorithm are shown in Tables 5.1 and 5.2 respectively. Both partitioning algorithms systematically achieve higher F-scores than the baseline, i.e., induce non-trivial clusterings and result in considerably higher purity, i.e., induce more adequate semantic roles.

For example, on the auto/auto dataset the agglomerative algorithm using cosine similarity increases F-score by 2.3 points over the baseline and by 7.2 points in terms of purity. This increase in purity is achieved by trading off against collocation, however in a favorable ratio as indicated by the overall higher F-scores.

While the scores of the two algorithms are often close to each other agglomerative par-

titioning systematically attains higher purity and F-score than label propagation. The latter trades off more purity and in return obtains higher collocation. Similar differences in F-score result from the different similarity functions with cosine similarity systematically outperforming avgmax similarity, confirming that cosine similarity is a more appropriate measure of cluster similarity for features where it is beneficial to capture the distributional similarity of clusters (see Section 5.3.0.0.4).

Table 5.3 shows the per-verb and per-role scores for the best-performing model on the auto/auto dataset, i.e., agglomerative partitioning using cosine similarity. The macroscopic results (higher F-score due to significantly higher purity) also hold pretty consistently across verbs and roles. An important exception is the verb *say* for which the baseline attains high scores due to only little variation in its syntactic realization within the corpus. While the model performs better on all core roles, there are some adjunct roles for which the baseline attains higher F-score. This is not surprising since the parser directly outputs certain labels such as *LOC* and *TMP* (see Appendix C) which results in high baseline scores for these roles.

Finally, Table 5.4 shows the 5 largest clusters output for the verb *Increase* for both the baseline and agglomerative partitioning using cosine similarity on the gold/gold dataset. For each cluster we output the 10 most frequent argument head lemmas. The special symbols REPLACED($) and REPLACED(CD) are those used as placeholders for monetary amounts and cardinal numbers respectively (see Section 3.2). In this case the model managed to induce an A0 cluster which is not present in the top 5 clusters of the baseline, although the cluster also incorrectly contains some A1 arguments which stem from a false merge. Generally, it is hard to notice a qualitative difference between the baseline and the model, which is not surprising given that scores are relatively close to each other and at a high level. The output for all the 12 selected verbs is given in Appendix D.

	Baseline			Agglomerative					
				avgmax			cosine		
	PU	CO	F1	PU	CO	F1	PU	CO	F1
auto/auto	68.3	**72.1**	70.1	75.3	69.2	72.1	**75.5**	69.5	**72.4**
gold/auto	74.9	**78.5**	76.6	80.3	73.8	76.9	**80.7**	74.0	**77.2**
auto/gold	77.0	71.5	74.1	84.9	70.8	77.2	**85.6**	**71.9**	**78.1**
gold/gold	81.6	**78.1**	79.8	87.4	75.3	80.9	**87.9**	75.6	**81.3**

Table 5.1: Results for agglomerative partitioning for both cosine and avgmax similarity on all datasets. All improvements over the baseline are statistically significant at level $\alpha < 0.001$ according to the test described in Appendix A.

	Baseline			Label Propagation					
				avgmax			cosine		
	PU	CO	F1	PU	CO	F1	PU	CO	F1
auto/auto	68.3	**72.1**	70.1	73.8	70.3	72.0	**74.0**	70.3	**72.1**
gold/auto	74.9	**78.5**	76.6	78.8	74.3	76.5	**79.2**	74.3	**76.7**
auto/gold	77.0	71.5	74.1	82.9	72.8	77.5	**83.6**	**73.1**	**78.0**
gold/gold	81.6	**78.1**	79.8	85.6	75.8	80.4	**86.3**	76.1	**80.9**

Table 5.2: Results for label propagation for both cosine and avgmax similarity on all datasets. All improvements over the baseline are statistically significant at level $\alpha < 0.001$ according to the test described in Appendix A.

5.6 Eager Similarity Combination

While other unsupervised learning problems in natural language processing have been addressed via graph partitioning (see the related work described in Section 5.7), we are not aware of other work which has employed multi-layer graphs. Rather, it is more common to use single-layer graphs, whose edge weights directly express instance-wise similarities. Such a graph can be obtained from the original multi-layer graph by collapsing the multiple feature layers into a single-layer graph as shown in Figure 5.4. Thereafter, the graph can be partitioned using the more standard label propagation algorithm for single-layer graphs given in Algorithm 5.

		Baseline			Agglomerative		
Verb	Freq	PU	CO	F1	PU	CO	F1
say	16698	**86.7**	**90.8**	**88.7**	85.8	90.4	88.0
make	4589	63.3	**71.0**	67.0	**66.4**	**71.0**	**68.6**
go	2331	47.3	**56.0**	51.3	**55.7**	55.3	**55.5**
increase	1425	58.0	69.0	63.0	**59.2**	**71.5**	**64.8**
know	1083	58.3	**70.8**	**63.9**	**58.6**	62.0	60.2
tell	969	59.0	**76.8**	66.7	**71.4**	68.0	**69.7**
consider	799	60.7	**65.3**	62.9	**71.0**	60.2	**65.1**
acquire	761	70.7	**78.4**	74.4	**72.0**	77.8	**74.8**
meet	616	70.0	**72.2**	71.1	**78.9**	68.3	**73.2**
send	515	68.3	**67.4**	67.9	**75.9**	64.9	**70.0**
open	528	55.3	**67.8**	**60.9**	**61.9**	55.1	58.3
break	274	51.1	**59.1**	54.8	**62.8**	55.8	**59.1**

(a) Per-verb scores.

		Baseline			Agglomerative		
Role	Freq	PU	CO	F1	PU	CO	F1
A0	49956	68.2	89.6	77.5	**71.1**	**90.0**	**79.4**
A1	72032	77.5	75.2	76.3	**80.7**	**76.9**	**78.7**
A2	16795	65.7	**71.4**	68.4	**79.1**	68.3	**73.3**
A3	2860	45.4	**81.8**	58.4	**71.7**	80.1	**75.7**
A4	2471	61.6	**86.1**	71.8	**81.6**	85.1	**83.3**
A5	44	46.4	59.1	52.0	**92.5**	**84.1**	**88.1**
AA	9	46.7	**100.0**	63.6	**50.0**	**100.0**	**66.7**
ADV	5824	33.8	**86.3**	48.6	**67.7**	41.9	**51.8**
CAU	878	67.5	**79.3**	72.9	**81.5**	73.9	**77.5**
DIR	811	51.5	**71.6**	59.9	**66.9**	58.9	**62.7**
DIS	3022	36.1	**90.4**	51.6	**57.5**	75.7	**65.3**
EXT	536	46.9	91.0	61.9	**70.2**	**92.2**	**79.7**
LOC	4481	65.1	**76.5**	**70.4**	**74.2**	58.4	65.3
MNR	5066	62.0	**64.6**	**63.3**	**84.3**	48.3	61.5
MOD	8064	80.2	44.1	56.9	**90.3**	**89.3**	**89.8**
NEG	2952	38.7	98.6	55.6	**53.5**	**98.7**	**69.4**
PNC	1682	67.9	**71.8**	69.8	**77.8**	70.6	**74.1**
PRD	56	39.1	**92.9**	55.1	**80.4**	85.7	**83.0**
REC	9	25.0	**100.0**	40.0	**75.0**	**100.0**	**85.7**
TMP	12928	71.1	**78.7**	**74.7**	**73.1**	43.1	54.2
NONE	49663	57.1	**47.3**	51.8	**71.6**	44.8	**55.1**

(b) Per-role scores.

Table 5.3: Fine-grained scores for Agglomerative Partitioning (cosine) on the auto/auto dataset.

Role	Examples
A1	it, sales, revenue, company, profit, rates, they, earnings, we, number
A1	number, reserves, stake, sales, costs, will, board, demand, rates, capacity
A4	REPLACED($), %, REPLACED(CD), yen, cent, #, member, earlier, kronor, years
ADV	REPLACED($), not, REPLACED(CD), also, be, increase, greatly, month, %, thus
A2	%, REPLACED($), REPLACED(CD), average, significantly, penny, yen, days, slightly, share

(a) Baseline

Role	Examples
A1	%, number, costs, sales, reserves, demand, stake, competition, pressure, size
A0	it, sales, revenue, company, profit, rates, earnings, we, they, line
A4	REPLACED($), %, REPLACED(CD), yen, cent, member, result, #, kronor, barrels
A3	REPLACED($), REPLACED(CD), %, yen, cent, earlier, period, #, member, quarter
TMP	year, quarter, month, years, period, september, REPLACED(CD), week, example, instance

(b) Agglomertive Clustering (cosine)

Table 5.4: Sample Output for the verb *Increase*. The output shows the 5 largest clusters and for each cluster the 10 most frequent argument head lemmas. The special symbols REPLACED($) and REPLACED(CD) are those used as placeholders for monetary amounts and cardinal numbers respectively (see Section 3.2).

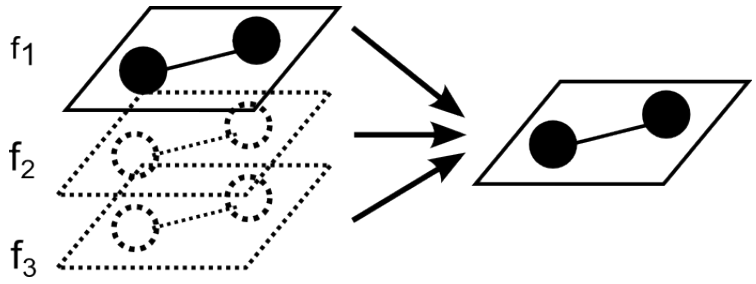


Figure 5.4: As an alternative approach the instance-wise similarity values for different features can be combined into a single similarity score between the instances. This is illustrated schematically in the figure above. The result of this aggregation is a single-layer graph which can then be divided into clusters using graph partitioning.

This approach differs from the one described in the previous section with respect to the order in which similarities are aggregated: whereas the two algorithms described so far firstly aggregate similarities on each feature layer and then combine them into an overall *cluster-wise* similarity score, the *eager* strategy considered here eagerly combines the feature similarities into an overall *instance-wise* similarity score.

In the following, we will consider both heuristically combining features similarities and a more principled approach in which the overall similarity function is estimated from a small amount of labeled training data, i.e., with weak supervision.

5.6.1 Label Propagation on Single-Layer Graphs

This section describes an algorithm for partitioning single-layer graphs, whose edges E directly quantify (overall) instance-wise similarities in contrast to the multi-layer graphs defined in Section 5.2.1, whose edges express similarities between instances *on a particular feature*. The algorithm is the single-layer version of the label propagation algorithm described in Section 5.4.

We assume each vertex v_i, here representing instances rather than atomic clusters, is assigned a label $l_i \in \{1 \dots L\}$ indicating the cluster it belongs to. Like for multi-layer

label propagation, each vertex initially belongs to its own cluster after which the algorithm proceeds by iteratively updating the label for each vertex, based on the labels of neighboring vertices:

$$l_i \leftarrow \arg\max_{l \in \{1...L\}} \sum_{v_j \in \mathcal{N}_i(l)} \phi(v_i, v_j) \tag{5.14}$$

Here, $\mathcal{N}_i(l) = \{v_j | (v_i, v_j) \in E \wedge l = l_j\}$ denotes the set of v_i's neighbors with label l. The algorithm is run for several iterations. At each iteration it passes over all vertices, and the update order of the vertices is chosen randomly.

5.6.1.0.12 Propagation Prioritization We make one important modification to the basic algorithm described so far based on the intuition that higher scores for a label indicate more reliable propagations. More precisely, when updating vertex v_i to label l we define the confidence of the update as the average similarity to neighbors with label l:

$$conf(l_i \leftarrow l) = \frac{1}{|\mathcal{N}_i(l)|} \sum_{v_j \in \mathcal{N}_i(l)} \phi(v_i, v_j) \tag{5.15}$$

We can then prioritize high-confidence updates by setting a threshold θ and allowing only updates with confidence greater or equal to θ. The threshold is initially set to 1 (i.e., the maximal possible confidence) and then lowered by a small constant $\Delta = 0.0025$ after each iteration until it reaches a minimum θ_{min}, at which point the algorithm terminates. This improves the resulting clustering, since it promotes reliable updates in earlier phases of the algorithm which in turn has a positive effect on successive updates.

5.6.2 Combining Feature Similarities Heuristically

One possibility is to heuristically combine feature similarity values into an overall similarity function, thereby relying on our prior knowledge about the problem. This constrains us to use only a small number of feature similarities whose relative influence on the overall similarity can be formulated explicitly: lexical similarity ϕ_{lex}, ϕ_{frame} which indicates occurrence in the same frame (also defined in Section 5.1.1) and syntactic similarity ϕ_{syn} which indicates whether two argument instances occur in a similar

Algorithm 5: Single-Layer Label Propagation Algorithm.

while *not done* **do**
  $A \leftarrow$ a list of all propagation graph vertices in a random order
  $i \leftarrow 0$
  while $i < length(A)$ **do**
    $l^* \leftarrow \arg\max_{l \in \{1 \ldots L\}} \sum_{v_j \in \mathcal{N}_i(l)} \phi(v_i, v_j)$
    $conf \leftarrow \frac{1}{|\mathcal{N}_i(l)|} \sum_{v_j \in \mathcal{N}_i(l)} \phi(v_i, v_j)$
    if $conf > \theta$ **then**
      $l_i \leftarrow l^*$
    end
    $i \leftarrow i + 1$
  end
  adjust thresholds
end

syntactic position. We define syntactic positions through the same four cues used in Section 5.3.0.0.2 for initialization: the relation of the argument head word to its governor, verb voice (active/passive), the linear position of the argument relative to the verb (left/right) and the preposition used for realizing the argument (if any). If the governor relation of the arguments is not the same the score is set to zero. Otherwise, the score is $\frac{S}{4}$ where S is the number of cues which agree, i.e., have the same value.

Based on these feature similarity functions we constructed an overall similarity function of the following form:

$$\phi(v_i, v_j) = \begin{cases} -\infty \text{ iff. } \phi_{frame}(v_i, v_j) = -1 \\ \lambda \phi_{lex}(v_i, v_j) + (1 - \lambda) \phi_{syn}(v_i, v_j) \text{ otherwise.} \end{cases} \tag{5.16}$$

The first case in the function constrains roles to be unique within a frame. Formally,

ϕ has range $ran(\phi) = [-1,1] \cup \{-\infty\}$ and for $x \in ran(\phi)$ we define $x + (-\infty) = -\infty$. This means that when summing over label scores a summand $-\infty$ results in an overall sum of $-\infty$, i.e., the propagation is ruled out. For the weighting parameter λ we chose a value $1/2$ based on our judgement that lexical and syntactic similarity are roughly of equal importance.

5.6.3 Learning Instance Similarities from Data

We can circumvent a heuristically chosen similarity function by using a (small) amount of labeled data to estimate an overall similarity function $\phi(v_i, v_j)$ between instance pairs, based on the values of M given feature-wise similarities $\phi_1(v_i, v_j), \ldots, \phi_M(v_i, v_j)$. For each pair of instances the overall similarity function $\phi(v_i, v_j)$ should indicate whether the semantic roles of the instances are the same $(+1)$ or not (-1). We are thus confronted with a classification problem, in which the overall similarity $\phi(v_i, v_j)$ corresponds to the classifiers' decision function whose value is determined by the individual feature similarities $\phi_f(v_i, v_j)$. Note, that in this setting we are no longer constrained to use only a small number of feature similarities, since now the influence of each feature similarity on the overall similarity is determined automatically.

In our experiments we used the support vector machine implemented by the SVM-Light package (Joachims, 1999) which is convenient since we can directly use as a similarity score the value of the decision function, normalized such that the maximal absolute value of the function is 1.

For training the classifier, we construct a training set by firstly sampling and labeling a set of L instances for a particular verb and then form all $L(L-1)$ possible pairs of instances. For each pair (v_i, v_j) we compute the feature similarity values $\phi_f(v_i, v_j)$ which inform the classification decision regarding the role-equality of the two instances. If the labels of the two instances agree, the overall similarity score (class value) is $+1$ and if they disagree the value is -1. We can repeat this for several verbs in order to obtain a (more) representative training sample. Specifically we sampled 100 instances for the five freely chosen verbs *say*, *go*, *increase*, *acquire*, *send* (i.e., we use a total of

	Baseline			Heuristic			Learned		
	PU	CO	F1	PU	CO	F1	PU	CO	F1
auto/auto	68.3	**72.1**	70.1	**70.1**	70.4	**70.2**	68.5	72.0	**70.2**
gold/auto	74.9	**78.5**	76.6	**76.4**	77.2	**76.8**	75.1	**78.5**	**76.8**
auto/gold	77.0	71.5	74.1	**79.6**	**72.6**	**75.9**	77.2	71.8	74.4
gold/gold	81.6	78.1	79.8	**83.7**	**78.2**	**80.9**	81.7	78.0	79.8

Table 5.5: Results obtained by label propagation when feature similarities between instances are combined eagerly into an overall similarity score. Column 'Heuristic' contains scores for the heuristically chosen overall similarity function (Section 5.6.2) and Column 'Learned' contains the scores for a similarity function that has been learned from data (Section 5.6.3)

.

500 labeled instances) which results in a training set of size 24750.

The trained classifier is applied to unlabeled instance pairs in order to determine their overall similarity values. We used a polynomial kernel of order 3 and incorporated the following similarity features: argument lemma, argument part of speech, frame the argument occurs in, governing relation, preposition used for argument realization, verb voice, subject part of speech, object part of speech and a feature that indicates whether the argument is directly attached to the verb.

5.6.4 Results and Analysis

The results obtained for both the heuristic and the learned similarity function are summarized in Table 5.5. Here we used a default value of $\theta_{min} = 0$ for the minimal confidence of a propagation as defined in Section 5.6.1.0.12. Contrary to our expectations, the learned similarity function does not outperform the baseline significantly, but instead leads to scores close to the baseline. We found that this due to the fact that the classifier makes similarity decisions mostly based on the feature that indicates whether the arguments occur in the same syntactic position and this in turn leads to the base-

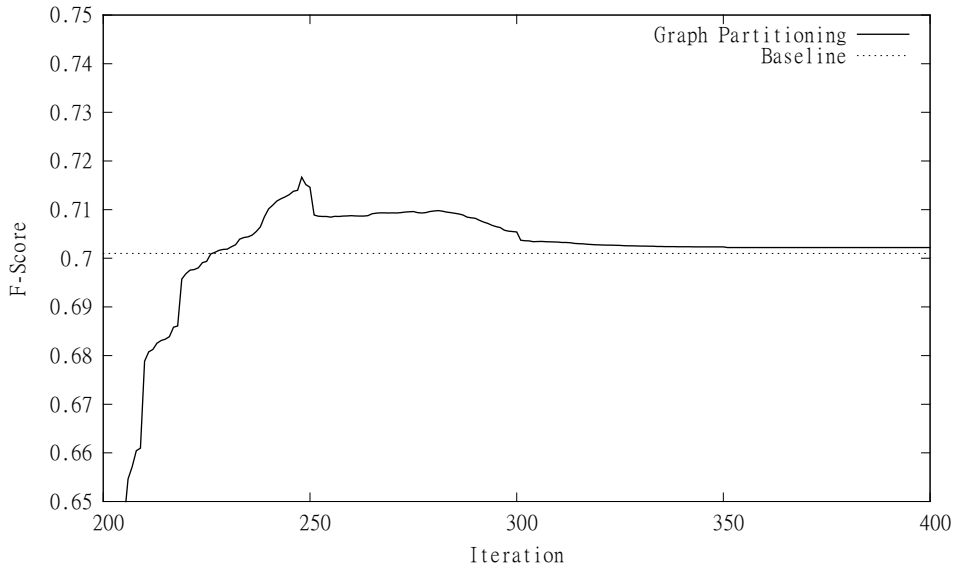


Figure 5.5: The F-score of the clusterings induced by Label Propagation using eager aggregation and the heuristically chosen similarity function for iterations 200 . . . 400 on the auto/auto dataset. The dotted line at 70.1 shows the baseline.

line solution (approximately). From a different perspective this again shows that the syntactic position is by far the most important feature for the task and the other similarity features provide (on average) relatively little extra information for determining the overall similarity.

Similarly, the heuristically chosen similarity function does not consistently result in scores much above the baseline. However, e.g., for the auto/auto dataset for which the final scores are roughly equal for the baseline, clusterings induced at intermediate stages of the algorithm actually attain higher F-score and purity than the baseline. This is illustrated in Figure 5.5 which shows the F-score of the induced clustering at each iteration on the auto/auto dataset. F-score rises above the baseline at iteration 227 and reaches its maximum of 71.7 at iteration 248 after which it drops again to its final value of 70.2. We could use a development set of labeled instances to determine an optimal stopping point, but like for the learned similarity function the method would then no longer be unsupervised.

5.7 Related Work

This section will discuss work which is related either in terms of the unsupervised semantic role induction *task* or in terms of the *methods* employed, or both.

Lin and Pantel (2001) cluster syntactic relations between pairs of words as expressed by parse tree paths into semantic relations by exploiting lexical distributional similarity. While their work is aimed at acquiring paraphrases in order to improve question answering it shares with this thesis the underlying idea of leveraging syntactic relations through lexical-semantic information.

Gamallo et al. (2005) cluster similar syntactic positions in order to develop models of selectional preferences which can be used for word sense induction and resolving attachment ambiguities. Thus while the objective of their work differs from the one here, there is a resemblance to the clustering methods described in this chapter, which also aim to group fine-grained syntactic positions into larger clusters, which in our case however represent semantic roles.

While graph partitioning has previously been applied to various problems in natural language processing (see Chen and Ji, 2010, for an overview) and other fields (Schaeffer, 2007), its application to semantic role induction is novel, as is the multi-layer approach. Our label propagation algorithm for graph partitioning was motivated by the Chinese Whispers algorithm proposed by Biemann (2006), also described in Abney (2007), pp. 146–147, under the name "clustering by propagation".

From an algorithmic perspective, information propagation on graphs is a general mechanism which has found its application within various formalisms, most notably as a means of inference within probabilistic graphical models (e.g., Yedidia et al., 2003; Minka, 2001). Closely related to our work is label propagation on similarity graphs for semi-supervised learning (Zhu et al., 2003). The main difference of our unsupervised method to these semi-supervised methods is that in our case none of the graph vertices (instances) are labeled with gold standard labels. For the semi-supervised setting it is therefore possible to define an empirically grounded objective function which penal-

izes label-disagreement on similar vertices, such as the original one used by Zhu et al. (2003):

$$min_{\hat{l}} \sum_{i,j} \phi(v_i, v_j) \left(1 - \mathbf{1}(l_i = l_j)\right)^2 \qquad \text{s.t. } \hat{l}_k = l_k \text{ if } l_k \text{ is labeled.} \tag{5.17}$$

Here, $\hat{l}$ is a vector containing the induced labels and l a vector containing the labels of labeled vertices[1] and pairwise similarities $\phi(v_i, v_j)$ are assumed to be non-negative. In this context, label propagation is more viewed as an algorithm for optimizing this objective function than as a graph partitioning algorithm. Nevertheless, the basic idea of transferring class membership information in a similarity graph is common to both settings and indeed the vertex update equations employed in the semi-supervised algorithms resemble those of the unsupervised algorithm and similarly employ an averaging procedure over label distributions of neighboring vertices (Talukdar, 2010). In the context of frame-semantic analysis semi-supervised label propagation has recently been applied in order to improve frame identification for unknown predicates (Das and Smith, 2011).

5.8 Summary

In this chapter we presented models which induce both adequate and non-trivial clusters of semantic roles. Our models are based on three linguistic principles:

1. Role-unambiguousness of syntactic positions within a specific linking;

2. Role-uniqueness within a frame;

3. Lexical-distributional equivalence of clusters representing the same semantic role.

The F-score for our models is higher than the baseline and purity is considerably higher. Especially in comparison to the relatively unsuccessful feature-based models

[1] For unlabeled vertices its entries are undefined.

described in Chapter 4 these results demonstrate the soundness of our similarity-driven models and the principles they are based on. Moreover, we argue that these principles are valid for all languages and thus our models are applicable to arbitrary languages, as will be discussed in the next chapter.

Finding adequate measures of similarity is a central issue within this approach and fundamentally more important than the particular choice of graph partitioning algorithm for cluster inference. In this respect, we have established the following key insight: while instance-wise similarity functions may be difficult to formulate and lead to unreliable computations, cluster-wise similarities can be computed reliably for sufficiently large clusters and based on theoretically sound principles, in our case, that clusters representing the same semantic role will have the same distribution over argument head words.

It is therefore also central to find a way of assigning instances to a set of initial clusters, such that these contain a sufficiently large number of instances. We have used the idea of initializing semantic role clusters by fine-grained syntactic position (i.e., a syntactic position within a particular linking) which has proven effective for the role induction task. Importantly, our models are most likely to induce increasingly accurate clusters as the size of the dataset is increased, although we leave an investigation of this point as future work.

A final point to emphasize is the conceptual transparency and clarity of the proposed approach. The principles underlying the data representation as a similarity-graph, the definition of similarities and the inference algorithms are immediately understandable and as such it is clear what exactly the models are inferring, and why so. In contrast to the models investigated in Chapter 4, there is no gap between the high-level modeling assumptions and the low-level inference mechanisms.

Chapter 6

Semantic Role Induction for German

So far we have solely induced semantic roles for English. However, the applicability of our models to arbitrary languages is important both from a theoretical and practical perspective. On one hand linguistic theory calls for models which are universal and not inherently coupled to any language-specific features. This is particularly true for models operating at the (frame-) semantic level, which arguably should be considered language-independent (Boas, 2005) despite of cross-lingual divergences in how frames are composed and realized (Padó, 2007). In any case, a model which would only work for one specific language could hardly be considered good. From a practical perspective, the benefit of a language-independent model is simply much greater, since it can be applied to arbitrary languages, genres and domains. Even if modifications in terms of parametrization or features are necessary the effort of applying an unsupervised model to a new, reasonably large dataset is of course smaller than manually labeling the instances it contains.

In this chapter we therefore assess the applicability of the models proposed in Chapter 5 to other languages, by examining how they perform on German. Although German as an Indo-European language is more closely related to English than for example Chinese, we nevertheless consider them sufficiently different to yield interesting conclusions. The most important differences with respect to frame semantic are the freer word order in German and extensive use of grammatical case marking, which will be

discussed in detail in Section 6.1. Moreover, since the underlying syntactic representations were developed independently we can realistically assess the capabilities of our models on operating on largely different syntactic representations.

As was mentioned in the previous chapter, we assume our models to be language-independent because they are based on a set of language-independent principles, namely *role-unambiguousness of syntactic positions within a specific linking*, *role-uniqueness within a frame* and *the distributional equivalence with respect to argument heads of clusters representing the same semantic role*. Moreover, we argue that it is possible to implement these principles in any language with an appropriate choice of features for defining syntactic positions and linkings, which may differ amongst languages. The results presented in this chapter supplement this theoretical argument and provide empirical evidence supporting it.

We start in Section 6.1 by discussing word order and case marking in German, which as was mentioned, differ from English and are closely related to frame semantics. Then in Section 6.2 we discuss the SALSA (Burchardt et al., 2006) dataset, on which we conduct our experiments. In Section 6.3 we describe the details of our model configuration and experimental setup. Results and their analysis are provided in Section 6.4.

6.1 Word Order and Case Marking in German

The high-level frame-semantic view of a German clause does not differ from that of an English clause: a frame is realized by a (verbal) predicate which binds together one or several nominal elements, each of which has a unique semantic role. However, there are fundamental differences in terms of how frame elements are mapped onto specific positions on the linear surface structure of a sentence, beyond the variation observed amongst verbs within a particular language. The following discussion of *word order* and *case marking* in German is based on S. Müller (2007).

Generally speaking, German places less constraints on word order (more precisely

phrase order) and instead possesses richer morphology that helps disambiguate the grammatical functions of linguistic units. In particular, the nominal arguments of a verb are marked with a grammatical case which directly indicates their grammatical function. Consider for example the following translations of the sentence *The landlord gave the key to the tenant*:

(6.1) [Der Vermieter]$_{NOM}$ gab [den Schlüssel]$_{ACC}$ [dem Mieter]$_{DAT}$.

(6.2) [Den Schlüssel]$_{ACC}$ gab [der Vermieter]$_{NOM}$ [dem Mieter]$_{DAT}$.

(6.3) [Dem Mieter]$_{DAT}$ gab [der Vermieter]$_{NOM}$ [den Schlüssel]$_{ACC}$.

The constituent *Der Vermieter* (*the landlord*) is marked with the *Nominative* case which identifies it as the *Subject* of the sentence, regardless of its linear position. Analogously, *den Schlüssel* (*the key*) holds the *Accusative* case that serves to identify the *Direct Object*, whereas *dem Mieter* (*the tenant*) is in the *Dative* case used for the *Indirect Object*. Since the grammatical function of these constituents is morphologically marked, the positioning of the arguments relative to the verb is freer for German than for English.

In fact, all six possible permutations of these three constituents form grammatical sentences and the positioning of arguments is primarily a stylistic element for achieving emphasis (typically the element before the verb is emphasized). The only constraint that applies here is that the verb *gab* must occur as the second element of the sentence. While in general, for main, declarative clauses the inflected verb part has to occur in second position, German is nevertheless commonly considered to be a verb-final language, as the verb (often) takes the final position in subordinate clauses, as do infinitive verbs (Brigitta, 1996).

In addition to the aforementioned cases (i.e., *Nominative*, *Accusative* and *Dative*) the *Genitive* case marks possession, much like the English possessive markers *of* and the apostrophe (’). It can also express various other semantic relations, e.g., properties of something, the source or goal of an action, etc. The *Genitive* therefore commonly

occurs within nominal constructions such as *die Stimme [des Volkes]$_{GEN}$* (*the voice of the people*), but certain verbs also license the *Genitive* for a particular argument:

(6.4) [Sie]$_{NOM}$ bedürfen [der dringenden Unterstützung]$_{GEN}$
They require immediate support.

(6.5) [Das Gericht]$_{NOM}$ beschuldigte [den Mann]$_{ACC}$ [der Steuerhinterziehung]$_{GEN}$
The court accused the man of tax evasion.

Note that in the second example, the subcategorization of the English verb *accuse* parallels that of its German counterpart, if in this context we take the preposition *of* to denote the English *Genitive* marker.

While prepositions are themselves considered case markers, case marking is also applied to the nominal parts of prepositional phrases. Often the particular case is simply licensed by the preposition and does not convey much additional information but there are situations where it directly serves to distinguish between semantic differences:

(6.6) Er sprang auf [**den** Tisch]$_{ACC}$.
*He jumped **onto** the table*

(6.7) Er sprang auf [**dem** Tisch]$_{DAT}$.
*He jumped **on** the table*

Here the *Accusative* indicates *Direction* and the *Dative* indicates *Location*. Note that the English translations of these sentences employ different prepositions to convey the different meanings.

While our models rely on a syntactic analysis which identifies the grammatical function of arguments, they do not directly model phrase order or syntactic case marking themselves. Therefore, the differences between English and German highlighted in this section do not pose a barrier that would prevent the application of our models to

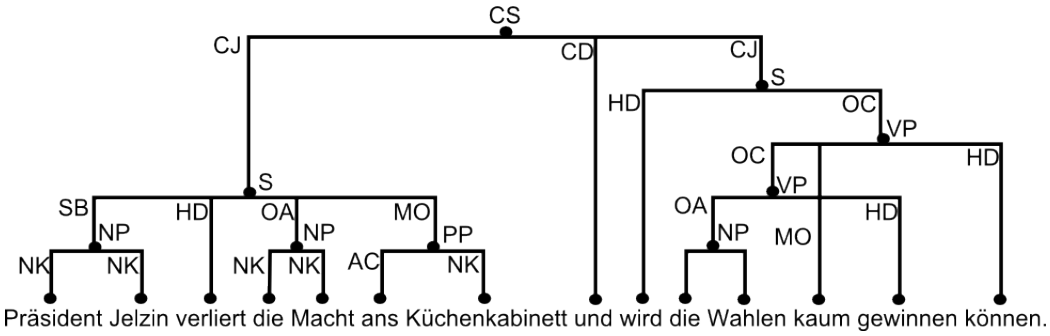


Figure 6.1: A sample parse tree for the (German) sentence *President Jelzin loses power to the kitchen cabinet and will hardly be able to win the elections*. The parse tree contains phrase labels *Noun Phrase* (NP), *Prepositional Phrase* (PP), *Verb Phrase* (VP), *Sentence* (S) and *Coordinated Sentence* (CS) and with dependency labels *Noun Kernel Element* (NK), *Subject* (SB), *Accusative Object* (OA), *Head* (HD), *Modifier* (MO), *Adpositional Case Marker* (AC), *Conjunct* (CJ) and *Clausal Object* (OC).

German and the models can in principle be informed by the same set of syntactic cues, as will be discussed in the following section.

6.2 The SALSA Dataset

SALSA (Burchardt et al., 2006) is a lexical resource for German, which like FrameNet for English, associates predicates with meaning representations in the form of frames. SALSA is built as an extra annotation layer over the TIGER corpus (Brants et al., 2002), a treebank for German consisting of around 40,000 sentences (700,000 tokens) of newspaper text, although to date not all predicate-argument structures have been annotated.

The frame and role inventory of SALSA was taken from FrameNet, but has been extended and adapted where necessary due to lack of coverage and cross-lingual divergences. The latter are linguistically interesting because they reveal differences in how languages convey the same situation (Padó and Erk, 2005). One instance of such a cross-lingual divergence described in Burchardt et al. (2006) is the occurrence of *Indi-*

rect Objects (*Dative Objects*) in German with no correspondences in English. Consider for example the following sentence and its word-for-word translation:

(6.8) *Er nahm* $[ihm]_{DAT}$ *das Bier aus der Hand.*
He took him the beer out of the hand.
He took the beer out of his hand

In the translation *He took the beer out of his hand*, there is no indirect object *him* taking the *Possessor* role, thereby unambiguously indicating that it is the hand of another person from which the *Subject* is taking the beer. Instead, the possessive is marked as part of the pronoun *his* and is therefore not considered to be a frame element of the predicate *Take* in FrameNet. In contrast, SALSA defines a *Possessor* role to mark such indirect objects, which is appropriate, given that they are realized as separate syntactic verbal arguments.

The syntactic structure of a sentence is represented through a constituent tree, whose terminal nodes represent individual tokens and whose non-terminal nodes represent phrases (see Figure 6.1). In addition to labeling each node with a constituent type such as *Sentence*, *Noun Phrase* and *Verb Phrase*, the edges between a parent and a child node are labeled according to the function of the child within the parent constituent, for example *Accusative Object*, *Noun Kernel Element* or *Head* (see Appendix C for a complete list of phrase and function labels). Edges can cross, "allowing local and non-local dependencies to be encoded in a uniform way and eliminating the need for traces. This approach has significant advantages for non-configurational languages such as German, which exhibit a rich inventory of discontinuous constituency types and considerable freedom with respect to word order" (Smith, 2003, p. 5). Compared to the Penn TreeBank (Marcus et al., 1993), tree structures are relatively flat. For example, the constituent structure does not encode whether a constituent is a verbal argument or adjunct, but instead this is encoded through the edge labels.

The frame annotations contained in SALSA do not cover all of the predicate-argument structures of the underlying TIGER corpus. Rather, only a subset of around 550 predicates with around 18,000 occurrences in the corpus have been annotated. Moreover, only core roles are annotated whereas adjunct roles are not, resulting in a smaller num-

ber arguments per predicate (1.96) compared to the CoNLL 2008 dataset (2.57) described in Section 3.2.

6.3 Experimental Setup

While the setup for the experiments in this chapter follows that described in Chapter 3, some deviations arose due to differences in the underlying dataset. Firstly, in contrast to the CoNLL 2008 dataset described in Section 3.2, the SALSA dataset (and the underlying TIGER corpus) does not supply automatic parse trees and we therefore conducted our experiments only on gold parses. Moreover, since adjunct arguments are not annotated in the SALSA dataset and because argument identification is not the central issue of this thesis we chose to also consider only the gold argument identification. Thus, all our experiments for German were carried out on the gold/gold dataset only.

A substantial linguistic difference between the German and English datasets is the sparsity of the argument head lemmas, which is significantly higher for German than for English: for the CoNLL 2008 dataset the average number of distinct head lemmas per verb is only 3.69, whereas for the SALSA dataset it is 20.12. This is partly due to the fact that the *Wall Street Journal* text underlying the English data is topically more focussed than the *Rundschau* newspaper text, which covers a broader range of news topics that are not limited to economics and politics. Moreover, noun compounding is more commonly employed in German than in English (see Corston-Oliver and Gamon, 2004), which leads to higher lexical sparsity.

This data sparsity affects our models, which crucially rely on lexical similarity for determining the role-equivalence of clusters. Therefore, we reduced the number of syntactic cues used for cluster initialization (see Section 5.3.0.0.2), in order to avoid creating too many small clusters, for which similarities cannot be reliably computed. Specifically, only the syntactic position and function word served as cues to initialize our clusters. Note that, like for English, the relatively small number of syntactic cues which determine the syntactic position within a linking is a consequence of the size

SB	OA	CJ	DA	CD	MO	RE	RS	OC	UC	OP	NK	CVC
16190	5631	5118	3293	1511	1262	925	406	257	218	216	143	84

Table 6.1: The counts of how many times a particular syntactic relation governs an argument in the dataset. Only relations with a count of greater than 80 are listed.

of our evaluation dataset (which is rather small) and not an inherent limitation of our models. On larger datasets, more syntactic cues could and should be incorporated in order to increase model performance.

No problem arises from the fact that SALSA follows the FrameNet annotation paradigm, in which several predicates can be associated with the same frame, whereas the CoNLL 2008 dataset contains verb-specific frames only. Since our models are designed to induce verb-specific frames, we convert SALSA frames into verb-specific (PropBank-like) frames by splitting each frame into several corresponding verb-specific frames and accordingly mapping frame roles onto verb-specific roles. We report per-verb scores for a selection of 10 verbs (seen in Table 6.3 a) which in some cases are translations of verbs used for English. For reporting per-role scores we however make use of the fact that roles have a common meaning across predicates (like e.g. A0 and A1 in PropBank), and report scores for a selection of 15 different roles (Table 6.3 b) with varied occurrence frequencies.

Our comparison will comprise agglomerative partitioning and the label propagation algorithm using both cosine- and avgmax-similarity as described in the previous chapter. We compare to the baseline described in Section 3.5. The model parameters α, β and γ which define the thresholds used in defining overall similarity scores were set and updated identically as described in Chapter 5, i.e., these parameters can be considered language and dataset-independent.

	German			English		
Model	PU	CO	F1	PU	CO	F1
Baseline	75.0	**81.7**	78.2	81.6	**78.1**	79.8
Agglomerative (avgmax)	**77.6**	80.8	**79.2**	87.3	75.3	80.9
Agglomerative (cosine)	**77.6**	80.8	**79.2**	**87.9**	75.6	**81.3**
Label Propagation (avgmax)	77.4	80.9	79.1	85.6	75.8	80.4
Label Propagation (cosine)	77.5	81.0	79.2	86.3	76.0	80.9

Table 6.2: Results of agglomerative partitioning and label propagation for both cosine and avgmax similarity on German. For comparison purposes results for English on the gold/gold dataset are also tabulated. All improvements over the baseline are statistically significant at level $\alpha < 0.001$ according to the test described in Appendix A.

6.4 Results and Analysis

The results of the baseline and our role induction models on the SALSA gold/gold dataset are shown in Table 6.2. For comparison purposes results for English on the gold/gold dataset are also tabulated. The baseline results in a similar F-score for both German and English, although the contributions of purity and collocation are different for the two languages. For English, purity is notably higher than for German whereas collocation is higher for German. This is not surprising, given the distribution over the syntactic relation that governs an argument given in Table 3.3 for English and Table 6.1 for German. For German, a few frequent labels absorb most of the probability mass, whereas for English the mass is distributed more evenly amongst the labels, leading to higher purity but lower collocation.

All four models attain scores close to each other and are both non-trivial and adequate. Like for English, the graph partitioning algorithms outperform the baseline in terms of F-score, although with around 1.0 points in F-score, the margin is lower for German than the best margin of 1.5 points for English. One reason is that the models incorporate less syntactic cues for initialization, due to the increased data sparsity described in the previous section. This also explains, why there is less spread between purity

		Baseline			Agglomerative (cosine)		
Verb	Freq	PU	CO	F1	PU	CO	F1
Sagen (*Say*)	2076	96.3	89.0	92.5	**97.3**	**97.7**	**97.5**
Wissen (*Know*)	487	79.7	76.0	77.8	**80.1**	**80.3**	**80.2**
Berichten (*Report*)	438	79.5	78.3	78.9	**80.0**	**81.3**	**80.7**
Nehmen (*Take*)	420	49.8	70.2	58.3	**51.9**	**72.4**	**60.5**
Verurteilen (*Convict*)	265	**70.9**	**83.4**	**76.7**	70.6	81.9	75.8
Erhöhen (*Increase*)	120	58.3	70.8	64.0	**70.8**	**73.3**	**72.1**
Schließen (*Close*)	93	40.9	72.0	52.1	**53.8**	**78.5**	**63.8**
Brechen (*Break*)	45	40.0	**91.1**	55.6	**44.4**	**91.1**	**59.7**
Schauen (*Watch*)	35	82.9	**91.4**	**86.9**	**85.7**	71.4	77.9
Plazieren (*Place*)	18	55.6	**83.3**	**66.7**	**66.7**	61.1	63.8
Treffen (*Meet*)	14	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Regnen (*Rain*)	12	66.7	**83.3**	**74.1**	**83.3**	50.0	62.5

(a) Per-verb scores.

		Baseline			Agglomerative (cosine)		
Role	Freq	PU	CO	F1	PU	CO	F1
Agent	1908	70.4	92.8	80.1	**70.5**	**93.9**	**80.5**
Theme	1637	69.1	79.2	73.8	**69.2**	**79.7**	**74.1**
Cognizer	1244	75.7	94.3	84.0	**76.2**	**94.6**	**84.4**
Entity	1195	**79.7**	85.9	**82.7**	78.6	**86.7**	82.4
Content	1136	87.2	65.2	74.6	**88.7**	**66.8**	**76.2**
Goal	1071	62.0	**81.0**	70.2	**87.0**	67.2	**75.9**
Topic	477	85.2	**69.4**	**76.5**	**86.8**	58.9	70.2
Source	267	**71.6**	**94.0**	**81.3**	66.1	76.0	70.7
Goods	171	73.0	**68.4**	**70.6**	**74.8**	66.7	70.5
Buyer	121	65.0	**90.1**	75.5	**70.4**	88.4	**78.4**
Employee	63	**50.4**	**98.4**	**66.7**	**50.4**	**98.4**	**66.7**
Required Situation	56	**60.3**	78.6	68.3	52.1	**82.1**	**63.8**
Opinion	50	66.7	50.0	57.1	**69.0**	**62.0**	**65.3**
Leader	29	**86.7**	**69.0**	**76.8**	**86.7**	65.5	74.6
Financed	25	79.3	**64.0**	70.8	**80.0**	**64.0**	**71.1**

(b) Per-role scores.

Table 6.3: Fine-grained scores for Agglomerative Partitioning (cosine) on German.

and collocation and the model scores are closer to the baseline scores than for English. However, qualitatively the tradeoff between purity and collocation is the same as for English, i.e., purity is increased at the cost of collocation.

Table 6.3 shows the per-verb and per-role scores for the best-performing model, i.e., agglomerative clustering using cosine similarity. The per-verb scores confirm, that data sparsity is affecting model performance, as can be seen from the fact that for the high-frequency verbs, which are less affected by the sparsity, scores are above the baseline whereas for lower-frequency verbs, this is not always the case. Analogously, the models tend to perform better on high-frequency roles, whereas there is no clear trend on lower-frequency roles. Like for English, it is difficult to identify qualitative differences between the output of the baseline and agglomerative partitioning given in Appendix D.

In contrast to English, for more than half of the verbs the models manage to outperform the baseline in terms of both purity *and* collocation, which is consistent with our macroscopic result, where the tradeoff between purity and collocation is not as strong as for English.

6.5 Summary

The experiments in this chapter have shown that our models can be successfully applied to languages other than English, thereby supporting the claim that they are based on a set of language-independent assumptions and principles. Despite of substantial differences between German and English grammar, both generally and in terms of the specific syntactic representation that was used, our models increased F-score over the baseline for both languages and resulted in a similar tradeoff between purity and collocation.

Confirming the conclusions from the previous chapter, data sparsity impedes the performance of our models. This was pronounced to the extent that we had to reduce the number of syntactic initialization cues in order to run the models on the relatively

small amount of gold-standard data. On larger datasets, more syntactic cues could be incorporated which together with the more reliable similarity estimates would most likely increase the performance of our models.

Chapter 7

Conclusions

Given the advances in parsing technology over the last two decades, frame-semantic analysis represents a realistic next step towards broad-coverage natural language understanding. The working hypothesis underlying this thesis has been that semantic roles can be induced without human supervision from a corpus of syntactically parsed sentences, by leveraging the syntactic relations conveyed through parse trees with lexical-semantic information. Thereby, we have challenged the established view that *supervised* learning is the method of choice for this task. We have argued that the shift to unsupervised methods is justified, given the fundamental problem of overcoming the lexical-semantic bottleneck, which under the supervised paradigm in the best case entails a massive human annotation effort and in the worst case may be practically infeasible. In the following we will summarize the main contributions of this thesis and discuss future work.

7.1 Contributions

1. We have conceptualized role induction in three different ways, corresponding to three largely differing underlying perspectives on the problem: once as probabilistic inference in a latent-variable model, once as determining the *canonical* syntactic posi-

tion of an argument, and once as a graph partitioning problem. As a whole the thesis therefore contributes towards a broader understanding of the role induction problem and frame-semantic analysis in general.

2. We have formulated and empirically validated a set of principles whose implementation allows a transition from a purely syntactic representation to a more semantic representation: (1) Role-unambiguousness of syntactic positions within a specific linking; (2) Role-uniqueness within a frame; (3) Lexical-distributional equivalence of clusters representing the same semantic role.

3. We introduced a novel *multi-layer* graph partitioning approach, that represents similarity between clusters on multiple feature layers, whose connectivity can be analyzed separately and then combined into an overall cluster-similarity score. We have demonstrated the superiority of this approach over the classical approach in which feature similarities are combined eagerly into instance-wise similarities.

4. We have contributed to the body of work on *similarity-driven* models, by demonstrating their suitability w.r.t. modeling our problem, their effectiveness, and their computational efficiency. The models are based on judgements regarding the similarity of argument instances with respect to their semantic roles. We showed that these judgements are comparatively simple to formulate and incorporate into a graph representation of the data.

A major advantage of our models is the immediateness with which the high-level knowledge guides the low-level inference procedure that is implemented by graph partitioning. The comparison with feature-based models (Chapter 4) reveals several advantages of the similarity-driven models and thereby provides a complementary view to much contemporary research which has concentrated on and argued in favor of feature-based models.

Our models are completely unsupervised and induce semantic roles solely from syntactic observations, whereby the number of induced roles is determined automatically. We have demonstrated the models' applicability to both English as well as German, applying identical parametrizations for both languages and without fundamentally changing

the underlying features, despite of the significant differences in the underlying syntactic representations.

We have demonstrated that these models consistently outperform the syntactic baseline across all datasets of automatic and gold parses, with gold and automatic argument identification and in English as well as German. The f-scores of our models are systematically above the baseline and the purity of induced clusters is considerably higher, although in most cases this increase in purity is achieved by decreasing collocation. In sum, these results provide strong empirical evidence towards the soundness of our models and the aforementioned principles they are based on.

5. We have identified major difficulties which arise and have provided analyses, which yield new insights into the problem of frame-semantic analysis and contribute towards developing better frame-based language understanding systems that are less reliant on labeled data, as will be discussed in the following section.

6. We have opened up a promising direction of research aimed at inducing shallow semantic representations without human supervision, which is a logical step given the relative maturity of syntactic parsing technology and the difficulty of overcoming the lexical-semantic bottleneck, i.e., the problem of acquiring large amounts of lexical-semantic knowledge.

7.2 Future Work

7.2.0.0.13 Data Sparsity Like for many other natural language processing problems, *data sparsity* poses a major barrier which affects both the feature-based models in Chapter 4 and the similarity-driven models in Chapter 5. There are two forms of data sparsity which arise in our context, namely the lexical sparsity of argument head lemmas and the sparsity of specific combinations of linking and syntactic position.

As our models are unsupervised, the conceptually simple solution to the data sparsity

problem is to train on larger datasets. Since our graph partitioning approaches could scale to larger datasets (in terms of orders of magnitude), this is an obvious next step. This would address both of the aforementioned forms of data sparsity. Firstly, it would allow us to incorporate a richer set of syntactic features for initialization and would therefore necessarily result in initial clusterings of higher purity. Secondly, the larger size of clusters would result in more reliable similarity scores. Augmenting the dataset would therefore almost surely increase the quality of induced clusterings.

7.2.0.0.14 Parser Reliance The reliance on a syntactic parser prohibits the application of our models to languages for which a parser is not available. Thus it would be potentially worthwhile though challenging to build models which operate on more readily available forms of syntactic analysis or even raw text. First steps in this direction have been taken by Abend and Rappoport (2010); Abend et al. (2009), who address unsupervised argument identification and core-adjunct distinction on the basis of part-of-speech tagged input which is subsequently analyzed by the unsupervised parser of Seginer (2007). Unfortunately, but not surprisingly, their method does not match the quality of a rule-based component which operates on parse trees produced by a supervised parser (see also Section 3.3).

Therefore, considering the notable difficulties of unsupervised parsing (Klein, 2005) entirely avoiding supervised parsers is probably not a realistic goal for the near future. Furthermore we argue, that although the unavailability of labeled training data also limits the applicability of supervised parsers across domains, genres and languages, the data requirements for parsing are probably not as extensive as for frame-semantic analysis. While syntax undoubtedly exhibits a considerable complexity and richness of possible constructions, these tend to be less tied to lexical idiosyncrasies and can therefore be learnt at a general rather than a predicate-specific level. Future research will have to determine, whether this indeed results in a decisive difference with respect to the feasibility of supervised learning for the tasks, i.e. whether parsing should and frame-semantic analysis should not be addressed with (fully) supervised learning.

7.2.0.0.15 Weak Supervision While interesting from a research perspective, the extreme of using no supervision at all does not seem appropriate for practical purposes. Applying *weak supervision* might help induce higher-quality semantic roles without sacrificing the benefit of only small human annotation effort. We have already discussed semi-supervised approaches in Section 2.3.2.0.5, in particular annotation projection between languages or within a single language (Fürstenau and Lapata, 2009; Padó and Lapata, 2009; van der Plas et al., 2011).

Alongside with finding suitable, possibly new models like annotation projection we think that addressing engineering issues would result in just as much if not more practical benefit. One such issue is sample selection, which should take into account the properties of the task. Specifically, simply labeling all sentences occurring in a corpus as was done for PropBank will not result in an optimal performance-effort ratio. For example for a verb like *say*, whose arguments can be labeled relatively accurately based solely on their syntactic position, it would presumably be possible to achieve good performance by labeling only a small fraction of all of the thousands of instances in the underlying Wall Street Journal corpus. A simple strategy for selecting relevant training samples could already help reduce the annotation effort. Alternatively, a more complex active learning strategy (see Tong, 2001), in which training samples are selected according to some optimality criterion could be applied.

7.2.0.0.16 Formalization and Probabilistic Modeling Finally, future research could aim at obtaining (more) formal results about problem and algorithms and possibly embedding our graph partitioning methods within a probabilistic framework or relating them to probabilistic graphical models. Future research on the problem would potentially profit, e.g., from a principled treatment of the various forms of uncertainty which accompany the problem and from a better understanding of the objective function which is being optimized by the graph partitioning algorithms.

As a possible starting point, consider the case of single layer graph partitioning, as described in 5.6. A single-layer similarity graph can be transformed into a probabilistic graphical model that specifies a distribution over vertex labels. In the transformed model each vertex corresponds to a random variable over labels and edges

are associated with binary potential functions over vertex-pairs. Let $\mathbf{1}(v_i = v_j)$ denote an indicator function which takes value 1 iff. $l_i = l_j$ and value 0, otherwise. Then pairwise potentials can be defined in terms of the original edge weights[1] as $\psi(v_i, v_j) = \exp(\mathbf{1}(v_i = v_j)\phi(v_i, v_j))$. A Gibbs sampler used to sample from the distribution of the resulting pairwise Markov random field (see Bishop, 2006; Wainwright and Jordan, 2008) employs almost the same procedure for updating a vertex label as the one in single-layer label propagation Equation 5.14, the difference being that labels would be sampled according to their probabilities, rather than chosen deterministically based on scores.

Label propagation algorithms are also commonly interpreted as random walks on graphs (see Talukdar, 2010). In our case such an interpretation is not directly possible due to the presence of negative edge weights, but this could be changed by transforming the edge weights onto a non-negative scale.

Yet another perspective arises by interpreting the update rule in Equation 5.14 as a heuristic for maximizing intra-cluster similarity and minimizing inter-cluster similarity. By assigning the label with maximal score to v_i, we greedily maximize the sum of intra-cluster edge weights while minimizing the sum of inter-cluster edge weights, i.e., the weight of the edge-cut. Cut-based methods in turn are also used for inference in pairwise Markov random fields like the one described above (Boykov et al., 2001).

Future work could consist of translating the multi-layer graph partitioning approach into one of these frameworks.

[1] Including weights with value zero and thus connecting all vertex pairs.

Bibliography

Abelson, R. and Schank, R. (1977). *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Abend, O. and Rappoport, A. (2010). Fully Unsupervised Core-Adjunct Argument Classification. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Abend, O., Reichart, R., and Rappoport, A. (2009). Unsupervised Argument Identification for Semantic Role Labeling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Abney, S. (2007). *Semisupervised Learning for Computational Linguistics*. Chapman & Hall/CRC.

Baker, C., Ellsworth, M., and Erk, K. (2007). SemEval-2007 Task 19: Frame Semantic Structure Extraction. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*.

Berg-Kirkpatrick, T., Bouchard-Côté, A., DeNero, J., and Klein, D. (2010). Painless Unsupervised Learning with Features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Berger, A., Pietra, S. D., and Pietra, V. D. (1996). A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics*, 22(1):39–71.

Biemann, C. (2006). Chinese Whispers: an efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*. Springer.

Boas, H. (2005). Semantic Frames as Interlingual Representations for Multilingual Lexical Databases. *International Journal of Lexicography*, 18(4):445–478.

Bottou, L. (2004). Stochastic Learning. In *Advanced Lectures on Machine Learning*, pages 146–168. Springer Verlag.

Boxwell, S., Mehay, D., and Brew, C. (2010). What a Parser Can Learn from a Semantic Role Labeler and Vice Versa. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories.*

Brigitta, H. (1996). Deutsch ist eine V/2-Sprache mit Verbendstellung und freier Wortfolge. In Lang, E. and Zifonun, G., editors, *Deutsch – typologisch*, pages 121–141. Walter de Gruyter.

Burchardt, A., Erk, K., Frank, A., Kowalski, A., Padó, S., and Pinkal, M. (2006). The SALSA Corpus: a German Corpus Resource for Lexical Semantics. In *Proceedings of the International Conference on Language Resources and Evaluation*.

Carreras, X. and Màrquez (2004). Introduction to the CoNLL-2004 Shared Task: Semantic Role Labeling. In *Proceedings of the Eighth Conference on Computational Natural Language Learning*.

Carreras, X. and Màrquez (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*.

Chambers, N. and Jurafsky, D. (2008). Unsupervised Learning of Narrative Event Chains. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Chambers, N. and Jurafsky, D. (2009). Unsupervised Learning of Narrative Schemas and Their Participants. In *Proceedings of the Joint Conference of the Annual Meet-*

ing of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP.

Chambers, N. and Jurafsky, D. (2011). Template-Based Information Extraction without the Templates. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Charniak, E. (1978). On the Use of Framed Knowledge in Language Comprehension. *Artificial Intelligence*, 11(3):225–265.

Chen, Z. and Ji, H. (2010). Graph-based clustering for computational linguistics: A survey. In *Proceedings of TextGraphs: The 2010 Workshop on Graph-based Methods for Natural Language Processing*.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press.

Christodoulopoulos, C., Goldwater, S., and Steedman, M. (2010). Two decades of unsupervised POS induction: How far have we come? In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Collins, M. and Koo, T. (2005). Discriminative Reranking for Natural Language Parsing. *Computational Linguistics*, 31(1):25–70.

Corston-Oliver, S. and Gamon, M. (2004). Normalizing German and English Inflectional Morphology to Improve Statistical Word Alignment. In Frederking, R. and Taylor, K., editors, *Machine Translation: From Real Users to Research*, volume 3265 of *Lecture Notes in Computer Science*, pages 48–57. Springer Berlin Heidelberg.

Cortes, C. and Vapnik, V. (1995). Support-vector Networks. *Machine Learning*, 20:273–297.

Das, D. and Smith, N. (2011). Semi-Supervised Frame-Semantic Parsing for Unknown Predicates. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38.

Dowty, D. (1991). Thematic Proto Roles and Argument Selection. *Language*, 67(3):547–619.

Fillmore, C. (1968). The Case for Case. In Bach, E. and Harms, R., editors, *Universals in Linguistic Theory*, pages 1–88. Holt, Rinehart and Winston, Inc., New York.

Fürstenau, H. and Lapata, M. (2009). Graph Aligment for Semi-Supervised Semantic Role Labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Gamallo, P., Agustini, A., and Lopes, G. (2005). Clustering Syntactic Positions with Similar Semantic Requirements. *Computational Linguistics*, 31(1):107–146.

Gerber, M. and Chai, J. (2010). Beyond NomBank: a Study of Implicit Arguments for Nominal Predicates. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Gildea, D. (2002). Probabilistic Models of Verb-Argument Structure. In *Proceedings of the International Conference on Computational Linguistics*.

Gildea, D. and Jurafsky, D. (2002). Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3):245–288.

Gordon, A. and Swanson, R. (2007). Generalizing Semantic Role Annotations Across Syntactically Similar Verbs. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Gordon, D. and Desjardins, M. (1995). Evaluation and Selection of Biases in Machine Learning. *Machine Learning*, 20:5–22.

Grenager, T. and Manning, C. (2006). Unsupervised Discovery of a Statistical Verb Lexicon. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Grosz, B., Weinstein, S., and Joshi, A. (1995). Centering: a Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*, 21:203–225.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., and Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the Conference on Computational Natural Language Learning: Shared Task*.

Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., and Tyson, M. (1997). FASTUS: A Cascaded Finite State Transducer for Extracting Information

from Natural Language Text. In Roche, E. and Schabes, Y., editors, *Finite State Language Processing*, pages 383–406. MIT Press.

Jain, A., Murty, M., and Flynn, P. (1999). Data Clustering: A Review. *ACM Computing Surveys*, 31(3).

Joachims, T. (1999). Making Large-Scale SVM Learning Practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

Johansson, R. and Moschitti, A. (2010). Syntactic and Semantic Structure for Opinion Expression Detection. In *Proceedings of the Conference on Computational Natural Language Learning*.

Johansson, R. and Nugues, P. (2008). Dependency Based Semantic Role Labeling of PropBank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Kamp, H. and Reyle, U. (1993). *From Discourse to Logic*, volume 1. Kluwer, Dordrecht.

Kaplan, R. and Bresnan, J. (1982). Lexical-Functional Grammar: A Formal System for Grammatical Representation. In Bresnan, J., editor, *The Mental Representation of Grammatical Relations*, pages 173–281. The MIT Press, Cambridge, MA.

Kipper, K., Dang, H. T., and Palmer, M. (2000). Class-Based Construction of a Verb Lexicon. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Klein, D. (2005). *The Unsupervised Learning of Natural Language Structure*. PhD thesis, Stanford University.

Klementiev, A. and Titov, I. (2011). A Bayesian Model for Unsupervised Semantic Parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Koomen, P., Punyakanok, V., Roth, D., and Yih, W. (2005). Generalized Inference with Multiple Semantic Role Labeling Systems. In *Proceedings of the Conference on Computational Natural Language Learning*.

Lang, J. and Lapata, M. (2010). Unsupervised Induction of Semantic Roles. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference*.

Lang, J. and Lapata, M. (2011a). Unsupervised Induction of Semantic Roles via Split-Merge Clustering. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Lang, J. and Lapata, M. (2011b). Unsupervised Semantic Role Induction with Graph Partitioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Levin, B. (1977). Mapping Sentences to Case Frames. Technical report, Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Working Paper Nr. 143.

Levin, B. (1993). *English Verb Classes and Alternations : a Preliminary Investigation*. The University of Chicago Press.

Levin, B. and Rappaport, M. (2005). *Argument Realization*. Cambridge University Press.

Levin, L. (1986). *Operations on Lexical Forms : Unaccusative Rules in Germanic Languages*. PhD thesis, Massachusetts Institute of Technology.

Lin, D. and Pantel, P. (2001). Discovery of Inference Rules for Question-answering. *Natural Langugae Engineering*, 7:343–360.

Litkowski, K. (2004). SENSEVAL-3 Task: Automatic Labeling of Semantic Roles. In *Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*.

Liu, D. and Nocedal, J. (1989). On the Limited Memory Method for Large Scale Optimization. *Mathematical Programming*, 45(3):503–528.

Mann, W. and Thompson, S. (1988). Rhetorical Structure Theory: Toward a Functional Theory of Text Organisation. *Text*, 3(8):234–281.

Manning, C., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Màrquez, L., Carrras, X., Litkowski, K., and Stevenson, S. (2008). Semantic Role La-

beling: an Introduction to the Special Issue. *Computational Linguistics*, 34(2):145–159.

Màrquez, L., Surdeanu, M., Comas, P., and Turmo, J. (2005). A Robust Combination Strategy for Semantic Role Labeling. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing*.

Melli, G., Wang, Y., Liu, Y., Kashani, M. M., Shi, Z., Gu, B., Sarkar, A., and Popowich, F. (2005). Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task. In *Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing Document Understanding Workshop*.

Merialdo, B. (1994). Tagging English Text with a Probabilistic Model. *Computational Linguistics*, 20(2).

Merlo, P. and Musillo, G. (2008). Semantic Parsing for High-precision Semantic Role Labelling. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*.

Merlo, P. and Stevenson, S. (2001). Automatic Verb Classification Based on Statistical Distributions of Argument Structure. *Computational Linguistics*, 27:373—-408.

Meyers, A., Reeves, R., Macleod, C., Szekely, R., Zielinska, V., Young, B., and Grishman, R. (2004). The NomBank Project: An Interim Report. In *Proceedings of the NAACL/HLT Workshop on Frontiers in Corpus Annotation*.

Miller, S., Stallard, D., Bobrow, R., and Schwartz, R. (1996). A Fully Statistical Approach to Natural Language Interfaces. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Minka, T. (2001). Expectation Propagation for Approximate Bayesian Inference. In *Proceedings of the Conference in Uncertainty in Artificial Intelligence*.

Minsky, M. (1974). A Framework for Representing Knowledge. Memo 306. Technical report, AI Laboratory, Massachusetts Institute of Technology.

Munkres, J. (1957). Algorithms for the Assignment and Transportation Problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38.

Padó, S. (2007). *Cross-Lingual Annotation Projection Models for Role-Semantic Information*. PhD thesis, Saarland University.

Padó, S. and Erk, K. (2005). To Cause Or Not To Cause: Cross-Lingual Semantic Matching for Paraphrase Modelling. In *Proceedings of the Workshop on Cross-Linguistic Knowledge Induction at EUROLAN*.

Padó, S. and Lapata, M. (2009). Cross-lingual Annotation Projection of Semantic Roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Palmer, M., Gildea, D., and Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

Parsons, T. (1994). *Events in the Semantics of English: a Study of Subatomic Semantics*. MIT Press.

Ponzetto, S. and Strube, M. (2006). Exploiting Semantic Role Labeling, WordNet and Wikipedia for Coreference Resolution. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Poon, H. and Domingos, P. (2009). Unsupervised Semantic Parsing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Pradhan, S., Ward, W., and Martin, J. (2008). Towards Robust Semantic Role Labeling. *Computational Linguistics*, 34(2):289–310.

Resnik, P. (1993). *Selection and Information: A Class-Based Approach to Lexical Relationships*. PhD thesis, University of Pennsylvania.

Riloff, E. and Schmelzenbach, M. (1998). An Empirical Approach to Conceptual Case Frame Acquisition. In *Proceedings of the Workshop on Very Large Corpora*.

Rosenberg, A. and Hirschberg, J. (2007). V-measure: A Conditional Entropy-based External Cluster Evaluation Measure. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C., and Scheffczyk, J. (2006). FrameNet II: Extended Theory and Practice, version 1.3. Technical report, International Computer Science Institute, Berkeley, CA, USA.

Ruppenhofer, J., Sporleder, C., Morante, R., Baker, C., and Palmer, M. (2010). SemEval-2010 Task 10: Linking Events and Their Participants in Discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*.

S. Müller (2007). *Head-Driven Phrase Structure Grammar: Eine Einführung*. Stauffenburg Verlag.

Schaeffer, S. (2007). Graph clustering. *Computer Science Review*, 1(1):27–64.

Schiller, A., Teufel, S., Stockert, C., and Thielen, C. (1999). Guidelines für das Tagging deutscher Textcorpora mit STTS. Technical report, University of Stuttgart and University of Tübingen.

Seginer, Y. (2007). Fast Unsupervised Incremental Parsing. In *Proceedings of the Annual Meeting of the Association of Computational Linguistics*.

Shen, D. and Lapata, M. (2007). Using Semantic Roles to Improve Question Answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Smith, G. (2003). A Brief Introduction to the TIGER Treebank, Version 1. Technical report, University of Potsdam.

Snyder, B., Naseem, T., and Barzilay, R. (2009). Unsupervised Multilingual Grammar Induction. In *Proceedings of the Conference of the Annual Meeting of the Association for Computational Linguistics*.

Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. (2003). Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

Surdeanu, M., Johansson, R., Meyers, A., and Màrquez, L. (2008). The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *Proceedings of the Conference on Natural Language Learning*.

Swanson, R. and Gordon, A. (2006). A Comparison of Alternative Parse Tree Paths for Labeling Semantic Roles. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*.

Swier, R. and Stevenson, S. (2004). Unsupervised Semantic Role Labelling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Talukdar, P. (2010). *Graph-Based Weakly Supervised Methods for Information Extraction & Integration*. PhD thesis, CIS Department, University of Pennsylvania.

Titov, I., Henderson, J., Merlo, P., and Musillo, G. (2009). Online Graph Planarisation for Synchronous Parsing of Semantic and Syntactic Dependencies. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*.

Tong, S. (2001). *Active Learning: Theory and Applications*. PhD thesis, Stanford University.

Toutanova, K., Haghighi, A., and Manning, C. (2008). A Global Joint Model for Semantic Role Labeling. *Computational Linguistics*, 34(2):161–191.

van der Plas, L., Merlo, P., and Henderson, J. (2011). Scaling up Automatic Cross-Lingual Semantic Role Annotation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Wainwright, M. and Jordan, M. (2008). Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305.

Wu, D. and Fung, P. (2009). Semantic Roles for SMT: A Hybrid Two-Pass Model. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*.

Xavier, L., Bott, S., and Màrquez, L. (2009). A Second-order Joint Eisner Model for Syntactic and Semantic Dependency Parsing. In *Proceedings of the Conference on Computational Natural Language Learning: Shared Task*.

Xue, N. and Palmer, M. (2004). Calibrating Features for Semantic Role Labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*.

Yarowsky, D. (1995). Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the Annual Meeting on Association for Computational Linguistics*.

Yedidia, J., Freeman, W., and Weiss, Y. (2003). Understanding Belief Propagation and its Generalizations. pages 239–269. Morgan Kaufmann Publishers Inc.

Zapirain, B., Agirre, E., Màrquez, L., and Surdeanu, M. (2010). Improving Seman-

tic Role Classification with Selectional Preferences. In *Proceedings of the North American Chapter of the Association for Computational Linguistics Conference*.

Zettlemoyer, L. and Collins, M. (2005). Learning to Map Sentences to Logical Form: Structured Classification with Probabilistic Categorial Grammar. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*.

Zhu, X., Ghahramani, Z., and Lafferty, J. (2003). Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. In *Proceedings of the International Conference on Machine Learning*.

Appendix A

Significance Testing

In order to compute the statistical significance of improvements over the baseline, we applied a sign test to a series of score pairs obtained by testing a particular method and the baseline on a subsample of the test data. Each subsample corresponds to a random selection of $M = 2000$ instances in the test set. We consider the resulting score pair samples to be 'sufficiently' independent to obtain indicative results from the test.

As null hypothesis we assume that

$$H_0 : \text{The method } m \text{ attains scores equal to the baseline } b.$$

Under H_0 the probability that method m outperforms the baseline b on a particular test set is $1/2$. Therefore the random variable S counting the number of times that $score_m > score_b$ in a sample of N score pairs is binomially distributed

$$S = \sum_{i=1}^{N} \mathbf{1}[score_m^{(i)} > score_b^{(i)}] \; Bin(1/2, N) \qquad .$$

We can therefore use S as our test statistic and reject H_0 if $S >> N/2$.

Appendix B

Argument Identification Rules

This appendix specifies the full set of relations used by rules (2) and (4) of the argument identification rules given for English in Section 3.3, Table 3.2. The symbols ↑ and ↓ denote the direction of the dependency relation (upward and downward,respectively). The dependency relations are explained in Table C.1 of Appendix C.

The relations in Rule (2) from Table 3.2 are IM↑↓, PRT↓, COORD↑↓, P↑↓, OBJ↑, PMOD↑, ADV↑, SUB↑↓, ROOT↑, TMP↑, SBJ↑, OPRD↑.

The relations in Rule (4) are ADV↑↓, AMOD↑↓, APPO↑↓, BNF↑↓-, CONJ↑↓, COORD↑↓, DIR↑↓, DTV↑↓-, EXT↑↓, EXTR↑↓, HMOD↑↓, IOBJ↑↓, LGS↑↓, LOC↑↓, MNR↑↓, NMOD↑↓, OBJ↑↓, OPRD↑↓, POSTHON↑↓, PRD↑↓, PRN↑↓, PRP↑↓, PRT↑↓, PUT↑↓, SBJ↑↓, SUB↑↓, SUFFIX↑↓ TMP↑↓, VOC↑↓ .

Appendix C

Label Sets

Table C.1: English dependency labels defined in Surdeanu et al. (2008), Table 4.

Label	Frequency	Description
NMOD	324834	Modifier of nominal
P	135260	Punctuation
PMOD	115988	Modifier of preposition
SBJ	89371	Subject
OBJ	66677	Object
ROOT	49178	Root
ADV	47379	General adverbial
NAME	41138	Name-internal link
VC	35250	Verb chain
COORD	31140	Coordination
DEP	29456	Unclassified
TMP	26305	Temporal adverbial or nominal modifier
CONJ	24522	Second conjunct (dependent on conjunction)
LOC	18500	Locative adverbial or nominal modifier
AMOD	17868	Modifier of adjective or adverbial
PRD	16265	Predicative complement
APPO	16163	Apposition
IM	16071	Infinitive verb (dependent on infinitive marker to)

HYPH	14073	Token part of a hyphenated word (dependent on a preceding part of the hyphenated word)
HMOD	13885	Token inside a hyphenated word (dependent on the head of the hyphenated word)
SUB	12995	Subordinated clause (dependent on subordinating conjuction)
OPRD	11707	Predicative complement of raising/control verb
SUFFIX	10548	Possessive suffix (dependent on possessor)
DIR	6145	Adverbial of direction
TITLE	5917	Title (dependent on name)
MNR	4753	Adverbial of manner
POSTHON	4377	Posthonorific modifier of nominal
PRP	4013	Adverbial of purpose or reason
PRT	3235	Particle (dependent on verb)
LGS	3115	Logical subject of a passive verb
EXT	2374	Adverbial of extent
PRN	2176	Parenthetical
EXTR	658	Extraposed element in cleft
DTV	496	Dative complement (to) in dative shift
PUT	271	Complement of the verb put
BNF	44	Benefactor complement (for) in dative shift
VOC	24	Vocative

Table C.2: The Penn Treebank part-of-speech tags for English defined in Marcus et al. (1993), Table 2.

Tag	Description
CC	coordinating conjunction
CD	cardinal number
DT	determiner
EX	existential *there*
FW	foreign word
IN	preposition or subordinating conjunction

JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
LS	list item marker
MD	modal
NN	noun, singular or mass
NNS	noun, plural
NNP	proper noun, singular
NNPS	proper noun, plural
PDT	predeterminer
POS	possessive ending
PRP	personal pronoun
PRP$	possessive pronoun
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	particle
TO	infinitival *to*
UH	interjection
VB	verb, base form
VBG	verb, gerund or present participle
VBN	verb, past participle
VBD	verb, past tense
VBP	verb, non-3rd person singular present
VBZ	verb, 3rd person singular present
WDT	wh-determiner
WP	wh-pronoun
WP$	possessive wh-pronoun
WRB	wh-adverb

Table C.3: The part-of-speech tags for German defined in Smith (2003) and based on Schiller et al. (1999).

Tag	Description
ADJA	adjective, attributive
ADJD	adjective, adverbial or predicative
ADV	adverb
APPR	preposition; circumposition left
APPRART	preposition with article
APPO	postposition
APZR	circumposition right
ART	definite or indefinite article
CARD	cardinal number
FM	foreign language material
ITJ	interjection
KOUI	subordinate conjunction
KOUS	subordinate conjunction
KON	coordinate conjunction
KOKOM	comparative conjunction
NN	common noun
NE	proper noun
PDS	substituting demonstrative pronoun
PDAT	attributive demonstrative pronoun
PIS	substituting indefinite pronoun
PIAT	attributive indefinite pronoun without determiner
PIDAT	attributive indefinite pronoun with determiner
PPER	non-reflexive personal pronoun
PPOSS	substituting possessive pronoun
PPOSAT	attributive possessive pronoun
PRELS	substituting relative pronoun
PRELAT	attributive relative pronoun
PRF	reflexive personal pronoun
PWS	substituting interrogative pronoun
PWAT	attributive interrogative pronoun

PWAV	adverbial interrogative or relative pronoun
PAV	pronominal adverb
PTKZU	'zu' before infinitive
PTKNEG	negative particle
PTKVZ	separable verbal particle
PTKANT	answer particle
PTKA	particle with adjective or adverb
SGML	SGML markup
SPELL	letter sequence
TRUNC	word remnant
VVFIN	finite verb, full
VVIMP	imperative, full
VVINF	infinitive, full
VVIZU	Infinitive with 'zu'
VVPP	perfect participle, full
VAFIN	finite verb, auxiliary
VAIMP	imperative, auxiliary
VAINF	infinitive, auxiliary
VAPP	perfect participle, auxiliary
VMFIN	finite verb, modal
VMINF	infinitive, modal
VMPP	perfect participle, modal
XY	non-word containing non-letter
$,	comma
$.	sentence-final punctuation mark
$(	other sentence-internal punctuation mark

Table C.4: The phrase labels for German defined in Smith (2003).

Tag	Description
AA	superlative phrase with *am*
AP	adjective phrase

AVP	adverbial phrase
CAC	coordinated adposition
CAP	coordinated adjective phrase
CAVP	coordinated adverbial phrase
CCP	coordinated complementiser
CH	chunk
CNP	coordinated noun phrase
CO	coordination
CPP	coordinated adpositional phrase
CS	coordinated sentence
CVP	coordinated verb phrase (non-finite)
CVZ	coordinated infinitive with *zu*
DL	discourse level constituent
ISU	idiosyncratic unit
MTA	multi-token adjective
NM	multi-token number
NP	noun phrase
PN	proper noun
PP	adpositional phrase
QL	quasi-language
S	sentence
VP	verb phrase (non-finite)
VZ	infinitive with *zu*

Table C.5: The dependency labels for German defined in Smith (2003).

Tag	Description
AC	adpositional case marker
ADC	adjective component
AG	genitive attribute
AMS	measure argument of adjective
APP	apposition

AVC	adverbial phrase component
CC	comparative complement
CD	coordinating conjunction
CJ	conjunct
CM	comparative conjunction
CP	complementizer
CVC	collocational verb construction (Funktionsverbgefüge)
DA	dative
DH	discourse-level head
DM	discourse marker
EP	expletive *es*
HD	head
JU	junctor
MNR	postnominal modifier
MO	modifier
NG	negation
NK	noun kernel element
NMC	numerical component
OA	accusative object
OA2	second accusative object
OC	clausal object
OG	genitive object
OP	prepositional object
PAR	parenthetical element
PD	predicate
PG	phrasal genitive
PH	placeholder
PM	morphological particle
PNC	proper noun component
RC	relative clause
RE	repeated element
RS	reported speech
SB	subject

SBP	passivised subject (PP)
SP	subject or predicate
SVP	separable verb prefix
UC	unit component
VO	vocative

Appendix D

Sample Output

The output below was generated by for a particular verb and model sampling the 5 largest clusters and for each of them sampling the 10 most frequent argument head lemmas. The special symbols REPLACED($) and REPLACED(CD) are those used as placeholders for monetary amounts and cardinal numbers respectively (see Section 3.2). For each cluster we indicate the majority gold standard role on the left. The output was generated on the gold/gold datasets. Since this output has not been manually edited, it contains lemmas such as –, which can be generated for example when the most frequent token of a proper noun is a hyphen, in which case it is chosen as the head (see Section 3.2).

Role	Examples
A0	mr., he, company, official, spokesman, analyst, trader, they, she, it
A1	be, have, will, would, do, expect, could, may, should, think
ADV	also, however, not, add, be, addition, still, refer, note, indeed
TMP	yesterday, now, week, month, friday, meanwhile, recently, then, later, year
LOC	statement, interview, filing, report, letter, conference, affidavit, meeting, testimony, here

(a) Baseline

Role	Examples
A1	be, have, will, would, do, but, expect, could, may, should
A0	mr., he, company, official, spokesman, analyst, trader, they, she, it
TMP	yesterday, month, now, week, friday, meanwhile, recently, example, then, year
DIS	also, however, not, still, indeed, only, separately, so, moreover, instead
LOC	statement, interview, filing, report, letter, affidavit, testimony, speech, sign, move

(b) Agglomertive Clustering (cosine)

Table D.1: Sample Output for the verb *say*.

Role	Examples
A1	it, them, offer, decision, sense, bid, money, product, payment, move
A0	it, he, they, mr., we, company, you, i, decision, that
A2	clear, difficult, available, possible, easy, work, hard, comparable, sure, REPLACED(CD)
ADV	not, also, of, just, have, only, be, accord, thus, even
TMP	REPLACED(CD), be, month, week, year, today, yesterday, years, time, never

(a) Baseline

Role	Examples
A1	it, decision, them, offer, sense, money, bid, product, payment, move
A0	it, he, they, mr., we, company, you, i, investor, that
A2	REPLACED(CD), clear, difficult, sure, available, possible, work, comparable, think, –
A0	company, industry, investment, investor, group, plant, loan, REPLACED(CD), unit, subsidiary
TMP	be, have, accord, month, today, yesterday, week, year, close, go

(b) Agglomertive Clustering (cosine)

Table D.2: Sample Output for the verb *make*.

Role	Examples
A1	you, he, it, we, they, %, market, i, company, price
ADV	not, ahead, probably, effect, even, just, too, back, also, bid
A4	REPLACED(CD), up, back, down, forward, out, way, away, in, market
TMP	now, REPLACED(CD), years, week, month, year, time, today, then, be
A1	will, would, way, price, 'll, step, things, those, could, can

(a) Baseline

Role	Examples
A1	you, it, he, we, they, price, company, market, i, %
A4	REPLACED(CD), market, effect, business, sale, level, work, offensive, college, detroit
NEG	not, now, then, probably, just, also, really, still, often, only
ADV	have, be, go, years, REPLACED($), bid, do, even, time, week
DIR	up, back, down, ahead, out, forward, away, further, in, too

(b) Agglomertive Clustering (cosine)

Table D.3: Sample Output for the verb *go*.

Role	Examples
A1	it, sales, revenue, company, profit, rates, they, earnings, we, number
A1	number, reserves, stake, sales, costs, will, board, demand, rates, capacity
A4	REPLACED($), %, REPLACED(CD), yen, cent, #, member, earlier, kronor, years
ADV	REPLACED($), not, REPLACED(CD), also, be, increase, greatly, month, %, thus
A2	%, REPLACED($), REPLACED(CD), average, significantly, penny, yen, days, slightly, share

(a) Baseline

Role	Examples
A1	%, number, costs, sales, reserves, demand, stake, competition, pressure, size
A0	it, sales, revenue, company, profit, rates, earnings, we, they, line
A4	REPLACED($), %, REPLACED(CD), yen, cent, member, result, #, kronor, barrels
A3	REPLACED($), REPLACED(CD), %, yen, cent, earlier, period, #, member, quarter
TMP	year, quarter, month, years, period, september, REPLACED(CD), week, example, instance

(b) Agglomertive Clustering (cosine)

Table D.4: Sample Output for the verb *increase*.

Role	Examples
A0	we, i, you, he, they, mr., it, investor, official, she
A1	be, will, have, that, it, mean, ’re, would, do, could
A2	not, also, even, “, really, REPLACED(CD), it, well, have, better
TMP	now, even, never, already, REPLACED(CD), days, always, disclose, today, sometimes
A1	people, company, anyone, critic, things, venture, puppy, doorman, somebody, wolfgang

(a) Baseline

Role	Examples
A0	we, i, you, he, they, mr., investor, it, official, she
A1	be, have, that, mean, ’re, it, could, ’ve, will, can
NEG	not, also, even, really, ever, apparently, only, then, widely, prior
A2	it, REPLACED(CD), “, mr., freeway, extent, he, newport, disclose, humulin
A1	people, company, all, something, anyone, someone, technology, incumbent, puppy, box

(b) Agglomertive Clustering (cosine)

Table D.5: Sample Output for the verb *know*.

Role	Examples
A2	be, you, us, have, would, him, reporter, them, do, will
A0	he, mr., i, they, we, she, you, investor, it, prosecutor
TMP	yesterday, week, month, never, recently, friday, wednesday, be, ever, years
ADV	not, just, also, ask, even, regulate, make, example, bug, instead
A1	buy, do, not, be, keep, make, choose, forget, pay, say

(a) Baseline

Role	Examples
A2	be, you, us, him, reporter, them, have, analyst, it, me
A0	he, mr., i, they, we, she, you, investor, it, prosecutor
A1	be, would, have, do, will, should, can, buy, could, expect
MOD	will, can, ca, could, may, would, must, ask, ’d, exactly
TMP	yesterday, week, friday, month, wednesday, be, tuesday, meet, night, monday

(b) Agglomertive Clustering (cosine)

Table D.6: Sample Output for the verb *tell*.

Role	Examples
A0	it, he, they, we, board, company, investor, group, you, i
A1	it, proposal, offer, himself, be, will, would, should, option, plan
A2	be, –, likely, seek, use, offer, problem, bid, investment, add
ADV	not, also, be, even, instance, example, say, widely, traditionally, generally
TMP	now, ever, time, week, still, longer, times, REPLACED(CD), monday, years

(a) Baseline

Role	Examples
A1	it, proposal, plan, offer, himself, option, bill, company, sale, alternative
A0	he, it, they, we, board, investor, company, director, group, i
A2	time, REPLACED(CD), –, signal, seek, use, offer, problem, bid, complete
A1	be, have, move, what, board, decide, feat, treasury, case, will
ADV	not, also, even, traditionally, generally, widely, officially, fully, instead, now

(b) Agglomertive Clustering (cosine)

Table D.7: Sample Output for the verb *consider*.

Role	Examples
A1	%, share, stake, business, company, unit, interest, assets, property, REPLACED(CD)
A0	it, company, they, he, mr., warner, bidder, american, unit, sony
A3	REPLACED($), dollar, not, stock, part, price, addition, be, itself, make
TMP	REPLACED(CD), year, ago, years, august, january, buy, go, month, then
A1	company, syndrome, group, unit, party, also, REPLACED($), eastern, lot, plan

(a) Baseline

Role	Examples
A1	%, share, company, stake, business, unit, interest, it, property, assets
A0	it, company, they, he, mr., bidder, daimler-benz, new, unit, warner
A3	REPLACED($), dollar, syndrome, stock, price, penny, c$, cash, combination, stake
A0	company, unit, transaction, group, party, purchase, giant, eastern, view, rest
A0	group, warner, pharmaceutical, unit, sony, state, management, first, broker, tenneco

(b) Agglomertive Clustering (cosine)

Table D.8: Sample Output for the verb *acquire*.

Role	Examples
A0	he, official, mr., they, it, company, bush, board, i, plan
A1	goal, demand, standard, requirement, target, payment, costs, resistance, redemption, REPLACED($)
A1	not, official, representative, mr., him, president, banks, worker, mediator, lunch
TMP	week, yesterday, REPLACED(CD), never, today, night, month, friday, tomorrow, meanwhile
LOC	chicago, new, office, washington, house, beijing, damascus, los, meeting, club

(a) Baseline

Role	Examples
A0	he, it, company, official, mr., they, bush, board, plan, i
A1	demand, requirement, goal, standard, target, payment, costs, resistance, redemption, deadline
TMP	week, REPLACED(CD), yesterday, today, friday, night, month, tomorrow, year, wednesday
A1	official, representative, mr., him, president, banks, mediator, worker, senator, treasury
PNC	discuss, consider, respond, try, make, outline, develop, determine, propose, resolve

(b) Agglomertive Clustering (cosine)

Table D.9: Sample Output for the verb *meet*.

Role	Examples
A1	signal, bill, share, letter, message, price, photo, them, newsletter, stock
A0	it, he, mr., congress, they, news, investor, bill, official, REPLACED(CD)
ADV	then, also, instead, even, tailspin, REPLACED($), demonstrator, relationship, up, sever
A2	back, REPLACED($), house, home, market, machine, culture, low, down, air
TMP	REPLACED(CD), week, then, yesterday, ago, day, complain, summer, month, session

(a) Baseline

Role	Examples
A1	bill, signal, message, share, letter, price, photo, it, market, newsletter
A0	it, he, mr., congress, they, –, news, computer, president, investor
A2	bush, REPLACED($), senate, subscriber, –, machine, another, mr., newspaper, los
A2	house, people, sheet, carrier, buying, appeal, facility, title, magazine, works
A2	soar, tumble, crash, low, billow, fall, fly, surge, nosedive, plunge

(b) Agglomertive Clustering (cosine)

Table D.10: Sample Output for the verb *send*.

Role	Examples
A0	it, currency, he, market, they, ual, stock, street, you, movie
A1	door, office, way, market, store, it, plant, REPLACED(CD), shop, account
A3	yen, trading, also, not, down, competition, probably, from, finally, be
TMP	REPLACED(CD), tokyo, monday, year, years, ago, first, friday, recently, week
LOC	tokyo, wall, west, air, venture, south, san, sudan, new, country

(a) Baseline

Role	Examples
A0	it, currency, he, market, they, ual, line, company, stock, mr.
A1	door, office, way, market, store, plant, REPLACED(CD), shop, account, –
TMP	REPLACED(CD), tokyo, monday, year, years, friday, follow, end, day, sometime
LOC	tokyo, bulgaria, wall, west, neb., france, venture, south, moscow, rotation
A3	competition, be, banks, world, public, bank, import, takeover, issuer, politics

(b) Agglomertive Clustering (cosine)

Table D.11: Sample Output for the verb *open*.

Role	Examples
A0	it, mr., unit, he, hell, story, REPLACED(CD), they, banks, unisys
A1	eggs, monopoly, –, law, ground, talks, will, streak, system, him
ADV	not, market, computer, quarter, try, also, blockade, match, month, then
TMP	be, year, quarter, now, days, flight, after, soon, eventually, morning
LOC	temblor, hotel, chiat, dark, higher, direction, japan, speech, that

(a) Baseline

Role	Examples
A0	it, mr., unit, he, hell, king, story, they, banks, unisys
A1	eggs, –, monopoly, talks, system, streak, him, ground, some, low
NEG	not, essentially, also, needlessly, nearly, about, then, however, neither
TMP	REPLACED(CD), quarter, vault, windows, month, mail, s&l
A1	market, computer, match, line, droplet, aspect, programming, riff

(b) Agglomertive Clustering (cosine)

Table D.12: Sample Output for the verb *break*.

Role	Examples
Speaker	er, Sprecher, sie, ich, vorsitzend, man, Scharping, wir, Präsident, Kohl
Message	sein, haben, werden, können, müssen, wollen, geben, sollen, stehen, dürfen
Medium	Gespräch, so, Rundfunk, Interview, Fernsehen, Deutschlandfunk, Journalist, Berlin, Rede, Landtag
Message	sein, haben, werden, müssen, kommen, liegen, wollen, finden, geben, brechen
Addressee	FR, Express, mir, Bild-Zeitung, ihm, Sonntagspost, Polizist, Focus, afp, Zeitung

(a) Baseline

Role	Examples
Speaker	er, Sprecher, sie, ich, vorsitzend, man, Scharping, wir, Präsident, Kohl
Message	sein, haben, werden, können, müssen, wollen, geben, der, sollen, stehen
Addressee	FR, Express, mir, Journalist, delegierter, Landtag, Bild-Zeitung, ihr, ihm, Kollege
Medium	Gespräch, Rundfunk, Interview, Fernsehen, Berlin, Rede, Deutschlandfunk, Bundestag , Begründung, ai-Interview
Manner	so, dazu, gut, anders, pathetisch, freiheraus, Bild, militärisch, wie, deshalb

(b) Agglomertive Clustering (cosine)

Table D.13: Sample Output for the verb *Sagen (Say)*.

Role	Examples
Cognizer	er, sie, wir, ich, man, der, niemand, wer, jeder, Leute
Content	sein, werden, haben, müssen, geben, können, sollen, wollen, tun, helfen
Content	es, der, nichts, wenig, Lösung, Antwort, Landgericht, US-Bürger, hervorbringen, Rat
Content	Plan, Mordplan, Sorge, Fall, Wert, Schritt, Politik, Menschenrechtsverletzung, Praxis, Mord-Absicht
Content	sein, wollen, haben, sie, Westen, Geld, Staat, unrichtig, fühlen, bedeuten

(a) Baseline

Role	Examples
Cognizer	er, sie, wir, ich, man, der, niemand, wer, jeder, Leute
Content	sein, werden, haben, müssen, wollen, können, geben, sollen, tun, sie
Content	es, der, nichts, wenig, Lösung, Antwort, Landgericht, US-Bürger, hervorbringen, Rat
Content	Plan, Mordplan, Sorge, Fall, Schritt, Menschenrechtsverletzung, Praxis, Frist, Mord-Absicht, Vorhaben
Content	wie, davon, nur, warum, so, übereinander, da

(b) Agglomertive Clustering (cosine)

Table D.14: Sample Output for the verb *Wissen (Know)*.

Role	Examples
Speaker	Fernsehen, FR, Rundfunk, Presse, Medium, Scientist, Journal, Post, Zeitung, Sender
Message	haben, sein, werden, erwägen, sollen, wollen, kommen, können, täuschen, dürfen
Message	wie, Ausgabe, Teil, Dezember-Ausgabe, ARD-Reportage, Wie, Vortrag, Treffen, Interview, Angriff
Topic	Prozeß, Wachstum, verletzter, Hausverbot, Handel, Rückgang, Wahlverlauf, Detail, Lust, Umsatz
Message	kommen, haben, sein, Bau, Seehofer, Schmeling, Hoechst, Sendeplatz, liegen, ziehen

(a) Baseline

Role	Examples
Speaker	Fernsehen, FR, Rundfunk, Presse, Medium, Scientist, Journal, Post, Zeitung, Sender
Message	haben, sein, werden, kommen, sollen, erwägen, wollen, liegen, ziehen, können
Message	wie, Wie
Topic	dieser, der, Prozeß, Wachstum, verletzter, Hausverbot, Handel, wer, Wahlverlauf, Militärberichterstatter
Medium	Ausgabe, Teil, Dezember-Ausgabe, Interview, Sondersendung, Telefongespräch, Rundfunk

(b) Agglomertive Clustering (cosine)

Table D.15: Sample Output for the verb *Berichten (Report)*.

Role	Examples
Agent	der, sie, man, er, wer, Polizei, Zahl, dieser, Frau, Staat
Theme	Stellung, Platz, Pille, Lauf, Geisel, wen, Abschied, Liverpooler, Zeit, der
Supported	ernst, Lupe, sich, Korn, Kenntnis, Hand, Markt, Titel, Auswahl, Pulle
Supported	Anspruch, Kenntnis, Visier, Pflicht, Untersuchungshaft, Feuer, Schutz, Angriff, Beschlag, Betrieb
Supported	Einfluß, Abschied, Rücksicht, Anleihe, Einblick, Ende, Trend, Anlauf, Geisel, Bezug

(a) Baseline

Role	Examples
Agent	der, sie, man, er, wer, Polizei, Zahl, dieser, Frau, Staat
Supported	Stellung, Abschied, Einfluß, Platz, Rücksicht, Geisel, Anleihe, der, Pille, Lauf
Supported	Anspruch, Visier, Pflicht, Untersuchungshaft, Schutz, Angriff , Beschlag, Betrieb, Besitz, Empfang
Supported	Kenntnis, Hand, Leitfigur, Mund, Auswahl, Gebiet, Maßstab, Vorbild, Schlepp, Gewahrsam
Source	sich, Korn, Titel, Zellentrakt, ANC-Mitglied, Haider, Stimme, Papst, Zweckbau, SPD

(b) Agglomertive Clustering (cosine)

Table D.16: Sample Output for the verb *Nehmen (Take)*.

Role	Examples
Defendant	er, der, sie, Gericht, Richter, Präsident, Scharping, angeklagter, Didier, Deckert
Charges	Mord, Rechtsbeugung, Fall, Kontakt, Nötigung, Schauprozeß, Beleidigung, Spionage
Finding	Haft, Tod, Haftstrafe, Geldstrafe, Gefängnis, Freiheitsstrafe, Jugendstrafe, Zahlung
Evaluee	Anschlag, Egoismus, Hinrichtung, Preisverleihung, Bestätigung, Sachsen-Anhalt
Judge	Staatssicherheitsgericht, angeklagter, Militärgericht, BGH, Nazi-Jurist, Junta

(a) Baseline

Role	Examples
Defendant	er, der, sie, Gericht, Richter, Präsident, Scharping, angeklagter, Braune, Didier
Evaluee	Anschlag, Egoismus, Ermordung, angeklagter, Hinrichtung, Staatssicherheitsgericht, Preisverleihung, Bestätigung, Sachsen-Anhalt, BGH
Finding	Haft, Tod, Haftstrafe, Gefängnis, Geldstrafe, Freiheitsstrafe, Jugendstrafe, Zahlung, Strafe, Todesstrafe
Charges	Mord, Rechtsbeugung, Kontakt, Nötigung, Beleidigung, Spionage, Volksverhetzung, Aufruhr, Terrorismus, Menschenrechtsverletzung
Case	Fall, Schauprozeß, Prozeß, Stiefelmord-Prozeß, Landesverrat, Telefongespräch, Aufruf, Sarajewo

(b) Agglomertive Clustering (cosine)

Table D.17: Sample Output for the verb *Verurteilen (Convict)*.

Role	Examples
Item	Zahl, sie, Investition, Arbeit, der, Wachstum, Instrument, Ausnutzung, Ausschüttung, Vergleichsquote
Value 2	REPLACED(CARD), Prozent, Mark, Million, Franc, Dollar, vierzehnfache, Tonne, Prozentpunkt, Leute
Item	Diskontsatz, Hochschulbau-Etat, Gefahr, Lohnnebenkosten, Beschäftigung, Kapital, Belegschaft, Wirksamkeit, Einkommen, Lebensqualität
Item	Ladestation, wahlberechtigter, Australien-Flug, Bett
Value 2	REPLACED(CARD), Milliarde

(a) Baseline

Role	Examples
Item	Zahl, sie, der, Investition, Arbeit, Wachstum, Instrument, Ausnutzung, Ausschüttung, Vergleichsquote
Item	Diskontsatz, Hochschulbau-Etat, Gefahr, Lohnnebenkosten, Beschäftigung, Kapital, Belegschaft, Wirksamkeit, Einkommen, Lebensqualität
Difference	REPLACED(CARD), Milliarde, Fünftel, Arbeitsplatz, Million, vierzehnfache, Tonne, Prozentpunkt, Pfennig, Ausländer
Value 2	Prozent, Mark, Franc, Dollar, Leute, REPLACED(CARD), Kalorie
Value 1	REPLACED(CARD), Million

(b) Agglomertive Clustering (cosine)

Table D.18: Sample Output for the verb *Erhöhen (Increase)*.

Role	Examples
Location	Schiffbauer, Ehe, Bundesbehörde, dieser, Guiskard, Kreis, Museum, Überblick, Geschäft, Beamter
Supported	Vertrag, Deckel, Landwirtschaftsschule, Vorvertrag, Frieden, Wehrübungsplatz, Friedensvertrag, Zedong, Filiale, Leutersdorf
Visitors	Schwelle, Kessel, Denkmodell, Synthese, Mark, verändern, Normalbürger, schwach, niedrig, Seite
Supported	Vertrag, übereinkommen, Pakt, Friedensabkommen, Friedensvertrag, Bündnis
Location	Apotheke, Fenster, Schule, Gedenkstätte, Park, stärken

(a) Baseline

Role	Examples
Location	Schiffbauer, Ehe, Bundesbehörde, dieser, Guiskard, Kreis, Museum, Überblick, Geschäft, Beamter
Supported	Vertrag, Friedensvertrag, Deckel, übereinkommen, Landwirtschaftsschule, Pakt, Vorvertrag, Frieden, Wehrübungsplatz, Zedong
Location	Fenster, Schule, Gedenkstätte, Park
Sub-event	Synthese, Bilanz, Mark
Visitors	Normalbürger, Öffentlichkeit, Luftverkehr

(b) Agglomertive Clustering (cosine)

Table D.19: Sample Output for the verb *Schließen (Close)*.

Role	Examples
Agent	er, Öko-Aktivisten, man, es, Achse, Meer, Verkäufer, terminieren, Yun, wir
Resistance	Vertraulichkeit, Protestaktion, Straßenblockade, Mehrheit, Genick, Knochen, Sieben-Milliarden-Rekord, derselbe, Realismus, Brückenbogen
Superregion	EU-Recht, Gestein, dir, Nebenabsprache
Relation	Empirismus, der
Victim	Entwicklung, ihnen

(a) Baseline

Role	Examples
Agent	er, Öko-Aktivisten, man, es, Achse, Meer, Verkäufer, terminieren, Yun, wir
Resistance	Vertraulichkeit, Protestaktion, Straßenblockade, Mehrheit, Genick, Knochen, Sieben-Milliarden-Rekord, derselbe, Realismus, Brückenbogen
Victim	Entwicklung, ihnen
Relation	Empirismus, der
Act	EU-Recht, Nebenabsprache

(b) Agglomertive Clustering (cosine)

Table D.20: Sample Output for the verb *Brechen (Break)*.

Role	Examples
Perceiver	sie, er, ich, Elitesoldat, Dupont, man, Deutschland, Sie, Agiv, Schwienbacher
Phenomenon	Geld, Rückspiegel, Glas, hinab, Stausee, Bonn, Gesicht, Röhre, Tempelanlage, Arbeitsmarkt
Controlled Entity	Vertragspartei, Beteiligung
Phenomenon	Film
Phenomenon	verwandeln

(a) Baseline

Role	Examples
Perceiver	sie, er, ich, Elitesoldat, Dupont, man, Deutschland, Sie, Agiv, Schwienbacher
Phenomenon	Geld, Tempelanlage, Arbeitsmarkt, HSV
Phenomenon	Rückspiegel, Glas, Gesicht, Röhre
Controlled Entity	Vertragspartei, Beteiligung
Phenomenon	sein

(b) Agglomertive Clustering (cosine)

Table D.21: Sample Output for the verb *Schauen (Watch)*.

Role	Examples
Goal	Privatanleger, Bank, Leitantrag, Amtskollege, Ausland, Szene, zentral
Capital	Anteil, er, Präsident, Mayer, mehr, Commerzbank
Theme	sich, Vorhang, zweiter, Rücktrittsforderung
Agent	Thierse

(a) Baseline

Role	Examples
Capital	Anteil, er, Präsident, Mayer, mehr, Commerzbank
Theme	sich, Vorhang, zweiter, Rücktrittsforderung
Investment	Privatanleger, Bank
Investment	Leitantrag, Ausland
Agent	Thierse

(b) Agglomertive Clustering (cosine)

Table D.22: Sample Output for the verb *Plazieren (Place)*.

Role	Examples
Theme	er, Bus, Detektiv, Flüchtling, Hussein, Marineeinheit, Berg, Kohl
Goal	Singapur, Südafrika, Dschenin, Stabsquartier, Travnik, Qingdao

(a) Baseline

Role	Examples
Theme	er, Bus, Detektiv, Flüchtling, Hussein, Marineeinheit, Berg, Kohl
Goal	Singapur, Südafrika, Dschenin, Stabsquartier, Travnik, Qingdao

(b) Agglomertive Clustering (cosine)

Table D.23: Sample Output for the verb *Treffen (Meet)*.

Role	Examples
Place	ständig, Frankfurt, Film, Tourquay, weniger, Nacht, einschwenken, Dänemark
Precipitation	Sternschnuppe, Schnee
Precipitation	Farbstoff
Time	wenn

(a) Baseline

Role	Examples
Place	Film, Tourquay, Nacht
Precipitation	Sternschnuppe, Schnee
Quantity	ständig, weniger
Place	Dänemark
Precipitation	Farbstoff

(b) Agglomertive Clustering (cosine)

Table D.24: Sample Output for the verb *Regnen (Rain)*.